

Oracle*9i* Application Server

Administrator's Guide

Release 2 (9.0.2)

May 2002

Part No. A92171-02

ORACLE®

Part No. A92171-02

Copyright © 2002 Oracle Corporation. All rights reserved.

The Programs (which include both the software and documentation) contain proprietary information of Oracle Corporation; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent and other intellectual and industrial property laws. Reverse engineering, disassembly or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. Oracle Corporation does not warrant that this document is error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Oracle Corporation.

If the Programs are delivered to the U.S. Government or anyone licensing or using the programs on behalf of the U.S. Government, the following notice is applicable:

Restricted Rights Notice Programs delivered subject to the DOD FAR Supplement are "commercial computer software" and use, duplication, and disclosure of the Programs, including documentation, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement. Otherwise, Programs delivered subject to the Federal Acquisition Regulations are "restricted computer software" and use, duplication, and disclosure of the Programs shall be subject to the restrictions in FAR 52.227-19, Commercial Computer Software - Restricted Rights (June, 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and Oracle Corporation disclaims liability for any damages caused by such use of the Programs.

Oracle is a registered trademark, and Oracle*MetaLink*, Oracle Store, Oracle9i, Oracle9iAS Discoverer, SQL*Plus, and PL/SQL are trademarks or registered trademarks of Oracle Corporation. Other names may be trademarks of their respective owners.

Contents

Send Us Your Comments	xiii
Preface.....	xv
Intended Audience	xvi
Documentation Accessibility	xvi
Organization.....	xvi
Related Documentation	xviii
Conventions.....	xix
1 Introduction to Application Server Administration	
Welcome.....	1-2
Priorities of an Application Server Administrator.....	1-2
Task 1: Determine Which Oracle9iAS Solutions You Require.....	1-3
Task 2: Design Your Application Server	1-3
Task 3: Install Your Application Server.....	1-3
Task 4: Learn to Use the Administration Tools.....	1-3
Task 5: Perform Postinstallation Tasks.....	1-4
Task 6: Configure Your Application Server.....	1-4
Task 7: Deploy Applications	1-4
Task 8: Monitor Your Application Server	1-4
Task 9: Back Up Your Application Server.....	1-4
Task 10: Reconfigure Your Application Server Design.....	1-5
What is in this Book?	1-5
Related Administration Documentation.....	1-6

2 Concepts for Administrators

Introduction	2-2
Application Server Installations	2-3
Application Server Install Types	2-3
About Configuring Components.....	2-6
The Oracle9iAS Infrastructure	2-7
When is an Infrastructure Required?.....	2-7
How Infrastructure Pieces Work Together	2-8
How Application Server Installations Use An Infrastructure.....	2-9
Configuring an Application Server Installation to Use an Infrastructure.....	2-9
Customer Databases	2-11
Components that Require a Customer Database	2-11
Components that Can Use a Metadata Repository or Customer Database	2-11
Depiction of a Component Using a Customer Database	2-12
Sample Application Server Enterprises.....	2-13
Designing an Application Server Enterprise.....	2-18
Step 1: Choose Your Application Server Install Types.....	2-18
Step 2: Choose Which Components to Configure.....	2-18
Step 3: Determine Your Infrastructure Requirements.....	2-19
Step 4: Choose Your Hosts	2-19
Step 5: Determine Your Customer Database Requirements.....	2-20
Step 6: Draw Your Application Enterprise Design.....	2-20
Installing an Application Server Enterprise.....	2-21
Oracle Homes	2-21
Coexistence With Other Products	2-21
Port Allocation	2-22
Primary and Secondary Installations.....	2-23
About Infrastructure Association.....	2-23
Managing an Application Server Enterprise	2-24
Application Server Instances.....	2-24
Component Instances.....	2-26
Farms	2-28
Clusters.....	2-29
Reconfiguring an Application Server Enterprise.....	2-29

3 Introduction to Administration Tools

Introduction	3-2
About the Oracle Enterprise Manager Web Site.....	3-2
Introducing the Enterprise Manager Home Pages	3-3
Getting Started with the Oracle Enterprise Manager Web Site	3-4
Starting and Stopping the Oracle Enterprise Manager Web Site	3-9
Changing the ias_admin Password	3-12
Enabling Enterprise Manager Web Site Security.....	3-13
About the Underlying Technologies.....	3-17
About the Oracle Enterprise Manager Console.....	3-18
The Enterprise Manager Three-Tier Framework	3-19
Installing the Framework	3-19
Starting and Stopping the Intelligent Agent on UNIX.....	3-21
Starting and Stopping the Intelligent Agent on Windows.....	3-21
Configuring the Management Server	3-21
Starting and Stopping Oracle Management Server on UNIX.....	3-24
Starting and Stopping the Oracle Management Server on Windows.....	3-25
Using the Console.....	3-25
Obtaining More Information	3-28
Using the Console with the Oracle Enterprise Manager Web Site	3-28

4 Postinstallation Tasks

Task 1: View the Oracle9iAS Welcome Page	4-2
Task 2: Check for Port Conflicts	4-3
Task 3: Complete Component Configuration	4-4
Task 4: Run Demos (optional)	4-7

5 Starting and Stopping the Application Server

Introduction	5-2
Starting an Application Server Enterprise	5-2
Stopping an Application Server Enterprise	5-2
Dependencies for Starting and Stopping Pieces of an Application Server Enterprise.....	5-3

Starting and Stopping Application Server Instances	5-4
About the Enterprise Manager Web Site and Intelligent Agent.....	5-4
Starting and Stopping Instances Using the Enterprise Manager Web Site	5-6
Starting and Stopping Instances Using Commands.....	5-7
Starting and Stopping Components.....	5-9
Starting and Stopping Components Using the Instance Home Page	5-9
Starting and Stopping Components Using the Component Home Page	5-9
Starting and Stopping Components Using Commands.....	5-10
About Components that Cannot Be Started or Stopped.....	5-12
Dependencies for Starting and Stopping Components.....	5-13
Starting an Infrastructure	5-15
About Unused Component Instances in the Infrastructure	5-15
Starting an Infrastructure Using Oracle Enterprise Manager on UNIX and Windows ...	5-17
Starting an Infrastructure Using Commands on UNIX	5-20
Starting an Infrastructure Using Commands on Windows.....	5-23
Stopping an Infrastructure	5-26
Stopping an Infrastructure Using Oracle Enterprise Manager on UNIX and Windows .	5-27
Stopping an Infrastructure Using Commands on UNIX	5-30
Stopping an Infrastructure Using Commands on Windows.....	5-32

6 Configuring HTTP Server, OC4J, and Web Cache

Introduction	6-2
About Manually Editing HTTP Server and OC4J Configuration Files.....	6-2
Configuring Oracle HTTP Server	6-3
Modifying the Document Root, Administrator E-mail, User, and Group Settings	6-4
Specifying a Port for a Listener.....	6-5
Changing the Error Log Properties	6-6
Adding an Access Log File	6-7
Changing the Access Log Properties	6-9
Managing the Client Request and Connection Handling.....	6-9
Editing the Server Configuration Files	6-10
Creating a Database Access Descriptor (DAD) for mod_plsql	6-11
Deleting a Database Access Descriptor (DAD) for mod_plsql	6-12

Enabling JServ	6-12
Directives Reference	6-13
Enabling JServ with mod_oprocmgr	6-16
Enabling JServ in Automatic Mode	6-17
Enabling JServ in Manual Mode	6-18
Using JServ and OC4J Together	6-19
Configuring Oracle9iAS Containers for J2EE	6-21
Creating an OC4J Instance	6-21
Deleting an OC4J Instance	6-22
Overview of OC4J Configuration Tasks	6-22
Configuring Oracle9iAS Web Cache	6-23

7 Component Configuration Dependencies

Oracle HTTP Server	7-2
Web Cache	7-7
OPMN	7-10
Portal	7-11
Web Providers (for Portals)	7-13
Provider Groups (for Portals)	7-15
Wireless	7-16
Oracle9iAS Single Sign-On	7-17
Oracle Internet Directory	7-21
Metadata Repository	7-24

8 Deploying Applications

Deploying J2EE Applications	8-2
Basic Deployment	8-3
Deploying the Petstore Web Application Demo	8-10
Downloading an OC4J-Ready Petstore Demo	8-10
Recovering From Deployment Errors	8-13
Undeploying J2EE Applications	8-13

9 Monitoring HTTP Server, OC4J, and Web Cache	
Introduction	9-2
Monitoring Oracle HTTP Server.....	9-2
Oracle HTTP Server Home Page	9-2
Status Metrics Page.....	9-3
Module Metrics Page.....	9-5
Response and Load Metrics Page	9-5
Monitoring Oracle9iAS Containers for J2EE.....	9-6
OC4J Home Page	9-6
Application Pages.....	9-9
Monitoring Oracle9iAS Web Cache	9-11
Oracle9iAS Web Cache Home Page.....	9-12
Activity Page	9-14
Application Web Servers Page.....	9-16
Popular Cached Documents Page	9-18
10 Reconfiguring the Application Server	
Configuring Additional Components	10-2
Extending an Application Server Installation.....	10-3
Adding an Additional Infrastructure.....	10-3
Associating an Instance with an Infrastructure (Joining a Farm).....	10-4
Changing the Schema Used by a Component	10-5
Changing Component Schema Passwords	10-5
11 Managing the Infrastructure	
Basic Infrastructure Administration	11-2
Starting and Stopping an Infrastructure.....	11-2
Backing up and Restoring an Infrastructure.....	11-3
Managing the Infrastructure Instance	11-3
Managing Oracle9iAS Single Sign-On.....	11-4
About Single Sign-On Installation and Configuration.....	11-5
About Single Sign-On Administrators	11-5
Accessing the Single Sign-On Server	11-6
Configuring Single Sign-On for Multilingual Support	11-6

Managing the Metadata Repository	11-7
About Metadata Repository Installation and Configuration	11-7
Starting and Stopping the Metadata Repository.....	11-8
Changing Metadata Repository Passwords.....	11-8
Relocating Metadata Repository Datafiles to a Different Directory	11-9
Managing Oracle Internet Directory.....	11-12
Starting and Stopping Oracle Internet Directory.....	11-12
Configuring Oracle Internet Directory and Single Sign-On on Separate Hosts.....	11-12
12 Managing Oracle Internet Directory	
 Oracle Internet Directory Security	12-2
About the Terms "Subscriber" and "Enterprise"	12-2
How Oracle9iAS Uses Oracle Internet Directory.....	12-3
New Oracle Internet Directory Instance or Existing Oracle Internet Directory Instance	12-3
Superuser and Administrators in Oracle Internet Directory	12-4
Relationship Between Oracle Internet Directory and the ias_admin User	12-5
Oracle9iAS Tree in Oracle Internet Directory	12-5
Tasks	12-6
Tools for Managing Oracle Internet Directory.....	12-7
 Delegating Administration Tasks.....	12-9
 Administering Users	12-11
Newly Created Oracle Internet Directory.....	12-11
Existing Oracle Internet Directory	12-11
Creating a User in Oracle Internet Directory.....	12-11
Adding a User to a Group	12-12
 Administering Groups.....	12-13
Newly Created Oracle Internet Directory.....	12-13
Existing Oracle Internet Directory	12-14
Group Details	12-14
To Determine a Group's Privileges.....	12-15
 Administering Privileges of Oracle9iAS Components	12-16
 Installing and Configuring Oracle9iAS.....	12-16
Newly Created Oracle Internet Directory.....	12-16
Existing Oracle Internet Directory	12-16
 Upgrading Oracle Internet Directory for Oracle9iAS	12-17

Password Policies	12-18
About Password Policies	12-18
Viewing the Password Policy	12-18
Modifying the Password Policy.....	12-19
Modifying Oracle Internet Directory Passwords	12-20
Modifying the orcladmin Password	12-20
Modifying a User Password.....	12-20
Changing Instance Passwords in Oracle Internet Directory	12-21
Removing an Application Server Instance From Oracle Internet Directory	12-22

13 High Availability

Introduction to High Availability	13-2
Benefits of High Availability.....	13-2
High Availability with Oracle9iAS Web Cache	13-3
Architecture	13-5
Oracle Process Management and Notification	13-6
Oracle Process Manager.....	13-6
Oracle Notification System.....	13-6
Configuring the Oracle Process Manager	13-7
process-manager	13-7
ohs	13-7
oc4j	13-8
custom	13-11
Sample Configurations for Oracle Process Manager.....	13-12
Configuring the Oracle Notification System	13-15
port.....	13-15
ssl.....	13-15
log-file.....	13-16
Sample Configuration Examples for Oracle Notification System	13-16
Failure Scenarios	13-17
Oracle HTTP Server Goes Down.....	13-17
OC4J Process Goes Down.....	13-19
Hi-AV Framework (OPMN) Goes Down.....	13-20

14 Application Server Clustering

About Oracle9iAS Clustering.....	14-2
Scalability.....	14-2
Availability	14-3
Manageability.....	14-4
Component Support.....	14-5
Non-Managed Clustering.....	14-5
Architecture.....	14-7
Front-End Load Balancer.....	14-8
Metadata Repository in the Infrastructure	14-9
Farm.....	14-9
Cluster	14-9
Application Server Instance	14-10
Management Features.....	14-12
Component Instances.....	14-14
J2EE Applications	14-20
Oracle Enterprise Manager Configuration Tree.....	14-21
Instance-Specific Parameters	14-22
Examples.....	14-22
Software Failure.....	14-23
Hardware Failure	14-24
State Replication	14-25
Cluster Configuration	14-26
Managing an Oracle9iAS Cluster.....	14-26
Managing Application Server Instances in a Cluster	14-29
OC4J Instance Configuration	14-31
Configuring Single Sign-On.....	14-35
Configuring Instance-Specific Parameters.....	14-37

A Component Descriptions

B Hardware Requirements

CPU, Disk, and TMP Requirements for Oracle9iAS	B-2
Memory Requirements for Oracle9iAS.....	B-3

C	Metadata Repository Schemas	
D	Default Port Numbers and Port Ranges	
	Oracle9iAS Port Usage (Sorted by Component)	D-2
	Oracle9iAS Port Usage (Sorted by Port Number).....	D-6
E	Configuration Files and Log Files	
	Configuration Files	E-2
	Log Files.....	E-7
F	DCM Command-Line Utility (dcmctl)	
	Overview.....	F-2
	About dcmctl Commands and Options.....	F-3
	Using dcmctl in a Clustered Environment.....	F-5
	Passing Parameters to the JVM.....	F-6
	Starting and Stopping	F-7
	Managing Application Server Instances	F-8
	Managing Components.....	F-9
	Managing Clusters.....	F-10
	Deploying Applications	F-12
	Saving a Backup	F-13
	Using the dcmctl Shell	F-14
	Executing dcmctl from a Command File	F-14
G	Enterprise Manager Command-Line Utility (emctl)	
H	Backup and Restore	
	Application Server and Infrastructure Oracle Homes.....	H-2
	Oracle Internet Directory.....	H-2
	Metadata Repository	H-2
	Customer Databases	H-2
I	Checking Oracle9iAS Version Numbers	
	Index	

Send Us Your Comments

Oracle9*i* Application Server Administrator's Guide, Release 2 (9.0.2)

Part No. A92171-02

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this document. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most?

If you find any errors or have any other suggestions for improvement, please indicate the document title and part number, and the chapter, section, and page number (if available). You can send comments to us in the following ways:

- Electronic mail: iasdocs_us@oracle.com
- FAX: 650-506-7407 Attn: Oracle9*i* Application Server Documentation Manager
- Postal service:
Oracle Corporation
Oracle9*i* Application Server Documentation
500 Oracle Parkway, M/S 2op3
Redwood Shores, CA 94065
USA

If you would like a reply, please give your name, address, telephone number, and (optionally) electronic mail address.

If you have problems with the software, please contact your local Oracle Support Services.

Preface

This preface contains these topics:

- Intended Audience
- Documentation Accessibility
- Organization
- Related Documentation
- Conventions

Intended Audience

Oracle9i Application Server Administrator's Guide is intended for anyone with an interest in Oracle9i Application Server or the latest Internet technologies.

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Standards will continue to evolve over time, and Oracle Corporation is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For additional information, visit the Oracle Accessibility Program Web site at

<http://www.oracle.com/accessibility/>

Accessibility of Links to External Web Sites in Documentation This documentation may contain links to Web sites of other companies or organizations that Oracle Corporation does not own or control. Oracle Corporation neither evaluates nor makes any representations regarding the accessibility of these Web sites.

Organization

This document contains:

Chapter 1, "Introduction to Application Server Administration"

This chapter serves as an introduction to application server administration and presents a prioritized approach to managing an application server.

Chapter 2, "Concepts for Administrators"

This chapter introduces concepts for designing, installing, and managing an application server.

Chapter 3, "Introduction to Administration Tools"

This chapter describes the Oracle9iAS Enterprise Manager Web site and the Oracle Enterprise Manager Console.

Chapter 4, "Postinstallation Tasks"

This chapter describes the setup tasks that must be performed after each application server installation.

Chapter 5, "Starting and Stopping the Application Server"

This chapter describes how to start and stop the application server.

Chapter 6, "Configuring HTTP Server, OC4J, and Web Cache"

This chapter describes common configuration tasks for three key components: Oracle HTTP Server, Oracle9iAS Containers for J2EE, and Oracle9iAS Web Cache.

Chapter 7, "Component Configuration Dependencies"

This chapter describes configuration parameter dependencies between components.

Chapter 8, "Deploying Applications"

This chapter provides steps for deploying applications on Oracle9iAS Containers for J2EE.

Chapter 9, "Monitoring HTTP Server, OC4J, and Web Cache"

This chapter discusses common monitoring tasks for three key components: Oracle HTTP Server, Oracle9iAS Containers for J2EE, and Oracle9iAS Web Cache.

Chapter 10, "Reconfiguring the Application Server"

This chapter describes tasks for reconfiguring the application server after initial setup, including adding installations, configuring components, and changing the database used by a component.

Chapter 11, "Managing the Infrastructure"

This chapter describes tasks for managing the pieces of the infrastructure, including Oracle9iAS Single Sign-On, Oracle Internet Directory, and the metadata repository.

Chapter 12, "Managing Oracle Internet Directory"

This chapter describes how to manage user privileges, group privileges, and passwords in Oracle Internet Directory.

Chapter 13, "High Availability"

This chapter introduces high availability concepts, and describes how to design and implement high availability for the application server.

Chapter 14, "Application Server Clustering"

This chapter introduces clustering concepts and describes how to implement and manage clustering.

Chapter A, "Component Descriptions"

This appendix provides a brief description of the components available in Oracle9iAS.

Chapter B, "Hardware Requirements"

This appendix provides the minimum hardware requirements for Oracle9iAS.

Chapter C, "Metadata Repository Schemas"

This appendix lists the Oracle9iAS metadata repository schemas.

Chapter D, "Default Port Numbers and Port Ranges"

This appendix lists the default port numbers and port ranges for Oracle9iAS components.

Chapter E, "Configuration Files and Log Files"

This appendix summarizes important configuration files and log files for Oracle9iAS.

Chapter F, "DCM Command-Line Utility (dcmctl)"

This appendix contains information about the DCM command-line utility (dcmctl).

Chapter G, "Enterprise Manager Command-Line Utility (emctl)"

This appendix contains information about the Enterprise Manager Web site command-line utility (emctl).

Chapter H, "Backup and Restore"

This appendix contains backup and recovery information for Oracle9iAS.

Chapter I, "Checking Oracle9iAS Version Numbers"

This appendix provides information about checking Oracle9iAS version numbers.

Related Documentation

For more information, see these Oracle resources:

- Oracle9i Application Server Documentation Library

- Oracle9*i* Application Server Platform-Specific Documentation on Oracle9*i* Application Server Disk 1

In North America, printed documentation is available for sale in the Oracle Store at

<http://oraclestore.oracle.com/>

Customers in Europe, the Middle East, and Africa (EMEA) can purchase documentation from

<http://www.oraclebookshop.com/>

Other customers can contact their Oracle representative to purchase printed documentation.

To download free release notes, installation documentation, white papers, or other collateral, please visit the Oracle Technology Network (OTN). You must register online before using OTN; registration is free and can be done at

<http://otn.oracle.com/admin/account/membership.html>

If you already have a username and password for OTN, then you can go directly to the documentation section of the OTN Web site at

<http://otn.oracle.com/docs/index.htm>

Conventions

This section describes the conventions used in the text and code examples of this documentation set. It describes:

- [Conventions in Text](#)
- [Conventions in Code Examples](#)
- [Conventions for Microsoft Windows Operating Systems](#)

Conventions in Text

We use various conventions in text to help you more quickly identify special terms. The following table describes those conventions and provides examples of their use.

Convention	Meaning	Example
Bold	Bold typeface indicates terms that are defined in the text or terms that appear in a glossary, or both.	When you specify this clause, you create an index-organized table .

Convention	Meaning	Example
<i>Italics</i>	Italic typeface indicates book titles or emphasis.	<p><i>Oracle9i Database Concepts</i></p> <p>Ensure that the recovery catalog and target database do <i>not</i> reside on the same disk.</p>
UPPERCASE monospace (fixed-width) font	Uppercase monospace typeface indicates elements supplied by the system. Such elements include parameters, privileges, datatypes, RMAN keywords, SQL keywords, SQL*Plus or utility commands, packages and methods, as well as system-supplied column names, database objects and structures, usernames, and roles.	<p>You can specify this clause only for a NUMBER column.</p> <p>You can back up the database by using the BACKUP command.</p> <p>Query the TABLE_NAME column in the USER_TABLES data dictionary view.</p> <p>Use the DBMS_STATS.GENERATE_STATS procedure.</p>
lowercase monospace (fixed-width) font	<p>Lowercase monospace typeface indicates executables, filenames, directory names, and sample user-supplied elements. Such elements include computer and database names, net service names, and connect identifiers, as well as user-supplied database objects and structures, column names, packages and classes, usernames and roles, program units, and parameter values.</p> <p>Note: Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.</p>	<p>Enter sqlplus to open SQL*Plus.</p> <p>The password is specified in the orapwd file.</p> <p>Back up the datafiles and control files in the /disk1/oracle/dbs directory.</p> <p>The department_id, department_name, and location_id columns are in the hr.departments table.</p> <p>Set the QUERY_REWRITE_ENABLED initialization parameter to true.</p> <p>Connect as oe user.</p> <p>The JRepUtil class implements these methods.</p>
lowercase italic monospace (fixed-width) font	Lowercase italic monospace font represents placeholders or variables.	<p>You can specify the parallel_clause.</p> <p>Run <code>Uold_release.SQL</code> where <code>old_release</code> refers to the release you installed prior to upgrading.</p>

Conventions in Code Examples

Code examples illustrate SQL, PL/SQL, SQL*Plus, or other command-line statements. They are displayed in a monospace (fixed-width) font and separated from normal text as shown in this example:

```
SELECT username FROM dba_users WHERE username = 'MIGRATE';
```

The following table describes typographic conventions used in code examples and provides examples of their use.

Convention	Meaning	Example
[]	Brackets enclose one or more optional items. Do not enter the brackets.	DECIMAL (<i>digits</i> [, <i>precision</i>])
{ }	Braces enclose two or more items, one of which is required. Do not enter the braces.	{ ENABLE DISABLE }
	A vertical bar represents a choice of two or more options within brackets or braces. Enter one of the options. Do not enter the vertical bar.	{ ENABLE DISABLE } [COMPRESS NOCOMPRESS]
...	Horizontal ellipsis points indicate either:	
	<ul style="list-style-type: none"> ■ That we have omitted parts of the code that are not directly related to the example ■ That you can repeat a portion of the code 	CREATE TABLE ... AS <i>subquery</i> ;
.	Vertical ellipsis points indicate that we have omitted several lines of code not directly related to the example.	SELECT <i>col1</i> , <i>col2</i> , ... , <i>coln</i> FROM employees;
.		
.		
Other notation	You must enter symbols other than brackets, braces, vertical bars, and ellipsis points as shown.	acctbal NUMBER(11,2); acct CONSTANT NUMBER(4) := 3;
<i>Italics</i>	Italicized text indicates placeholders or variables for which you must supply particular values.	CONNECT SYSTEM/ <i>system_password</i> DB_NAME = <i>database_name</i>
UPPERCASE	Uppercase typeface indicates elements supplied by the system. We show these terms in uppercase in order to distinguish them from terms you define. Unless terms appear in brackets, enter them in the order and with the spelling shown. However, because these terms are not case sensitive, you can enter them in lowercase.	SELECT last_name, employee_id FROM employees; SELECT * FROM USER_TABLES; DROP TABLE hr.employees;

Convention	Meaning	Example
lowercase	<p>Lowercase typeface indicates programmatic elements that you supply. For example, lowercase indicates names of tables, columns, or files.</p> <p>Note: Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.</p>	<pre>SELECT last_name, employee_id FROM employees; sqlplus hr/hr CREATE USER mjones IDENTIFIED BY ty3MU9;</pre>

Conventions for Microsoft Windows Operating Systems

The following table describes conventions for Microsoft Windows operating systems and provides examples of their use.

Convention	Meaning	Example
Choose Start >	How to start a program.	To start the Oracle Database Configuration Assistant, choose Start > Programs > Oracle - HOME_NAME > Configuration and Migration Tools > Database Configuration Assistant.
File and directory names	File and directory names are not case sensitive. The following special characters are not allowed: left angle bracket (<), right angle bracket (>), colon (:), double quotation marks ("), slash (/), pipe (), and dash (-). The special character backslash (\) is treated as an element separator, even when it appears in quotes. If the file name begins with \\, then Windows assumes it uses the Universal Naming Convention.	c:\winnt\"system32 is the same as C:\WINNT\SYSTEM32
C:\>	Represents the Windows command prompt of the current hard disk drive. The escape character in a command prompt is the caret (^). Your prompt reflects the subdirectory in which you are working. Referred to as the <i>command prompt</i> in this manual.	C:\oracle\oradata>

Convention	Meaning	Example
	The backslash (\) special character is sometimes required as an escape character for the double quotation mark ("") special character at the Windows command prompt. Parentheses and the single quotation mark ('') do not require an escape character. Refer to your Windows operating system documentation for more information on escape and special characters.	C:\>exp scott/tiger TABLES=emp QUERY=\ "WHERE job='SALESMAN' and sal<1600\" C:\>imp SYSTEM/ <i>password</i> FROMUSER=scott TABLES=(emp, dept)
<i>HOME_NAME</i>	Represents the Oracle home name. The home name can be up to 16 alphanumeric characters. The only special character allowed in the home name is the underscore.	C:\> net start Oracle <i>HOME_NAME</i> tnsListener

Convention	Meaning	Example
<i>ORACLE_HOME</i> and <i>ORACLE_BASE</i>	<p>In releases prior to Oracle8<i>i</i> release 8.1.3, when you installed Oracle components, all subdirectories were located under a top level <i>ORACLE_HOME</i> directory that by default used one of the following names:</p> <ul style="list-style-type: none"> ■ C:\orant for Windows NT ■ C:\orawin95 for Windows 95 ■ C:\orawin98 for Windows 98 	<p>Go to the <i>ORACLE_BASE\ORACLE_HOME\rdbms\admin</i> directory.</p>

Introduction to Application Server Administration

This chapter provides an introduction to Oracle9*i* Application Server administration and an overview of the responsibilities of an application server administrator.

It contains the following topics:

- [Welcome](#)
- [Priorities of an Application Server Administrator](#)
- [What is in this Book?](#)
- [Related Administration Documentation](#)

Welcome

Welcome to Oracle9*i* Application Server!

Oracle9*i* Application Server (Oracle9*i*AS) is a powerful and feature-rich platform that supports all types of Web applications and Internet standards. Whether your application server hosts a collaborative Web site, a corporate portal, or an expansive e-business, your role as an application server administrator is the same: to provide a secure, available, and convenient platform for application deployers and application users.

This book provides an introduction to application server administration and the Oracle9*i*AS solutions and features that will enable you to provide a reliable and efficient application server that meets the demands of your user community.

Priorities of an Application Server Administrator

The following tasks present a prioritized approach to managing Oracle9*i* Application Server:

- [Task 1: Determine Which Oracle9*i*AS Solutions You Require](#)
- [Task 2: Design Your Application Server](#)
- [Task 3: Install Your Application Server](#)
- [Task 4: Learn to Use the Administration Tools](#)
- [Task 5: Perform Postinstallation Tasks](#)
- [Task 6: Configure Your Application Server](#)
- [Task 7: Deploy Applications](#)
- [Task 8: Monitor Your Application Server](#)
- [Task 9: Back Up Your Application Server](#)
- [Task 10: Reconfigure Your Application Server Design](#)

Task 1: Determine Which Oracle9iAS Solutions You Require

Oracle9iAS offers a variety of solutions you can use to build your application server. These solutions include J2EE and other industry-standard application support, portals, wireless support, caching, business intelligence features, security features, and management tools. Your first task as an administrator is to determine which Oracle9iAS solutions your user community requires.

See Also: *Oracle9i Application Server Concepts Guide* for more information about Oracle9iAS solutions

Task 2: Design Your Application Server

Designing an application server involves determining which Oracle9iAS features to install and configure, which hosts to use, and which databases (if any) are required to support your application server.

See Also: [Chapter 2, "Concepts for Administrators"](#) for an introduction to application server design

Task 3: Install Your Application Server

You can install your application server using Oracle Universal Installer, an easy-to-use tool that automatically installs and configures your application server.

See Also: *Oracle9i Application Server Installation Guide* and the Oracle documentation specific to your operating system

Task 4: Learn to Use the Administration Tools

Oracle9iAS provides a Web-based tool, a Java-based console, and command-line utilities for performing management tasks, including starting and stopping the application server, configuring the application server, deploying applications, and monitoring the application server. These tools provide a centralized interface for managing application servers, as well as the databases in your enterprise.

See Also: [Chapter 3, "Introduction to Administration Tools"](#)

Task 5: Perform Postinstallation Tasks

You must perform some setup tasks after every application server installation. These tasks include checking port allocation and configuring certain application server features to use custom-tuned databases.

See Also: [Chapter 4, "Postinstallation Tasks"](#)

Task 6: Configure Your Application Server

Oracle9iAS is a very flexible platform that offers many options to customize its services. You can change passwords, add functionality, and configure a variety of parameters including security settings, resource limits, logging features, and user privileges.

See Also: [Chapter 6, "Configuring HTTP Server, OC4J, and Web Cache"](#)

Task 7: Deploy Applications

You can easily deploy and redeploy J2EE applications on Oracle9iAS using a Web-based wizard that guides you through configuring, deploying, and securing your application, as well as publishing its Web Services.

See Also: [Chapter 8, "Deploying Applications"](#)

Task 8: Monitor Your Application Server

It is important to monitor your application server to identify configuration problems and performance bottlenecks. Oracle9iAS offers centralized, graphical monitoring tools that characterize the performance of every aspect of your application server.

See Also: [Chapter 9, "Monitoring HTTP Server, OC4J, and Web Cache"](#)

Task 9: Back Up Your Application Server

You should back up configuration files, log files, applications, and data on a regular basis. You can back up and restore Oracle9iAS using standard file system and database backup tools and strategies.

See Also: [Appendix H, "Backup and Restore"](#)

Task 10: Reconfigure Your Application Server Design

Oracle9iAS is a scalable and flexible product that allows you to expand and reconfigure your original application server design to meet changing user requirements and increasing load.

See Also: [Chapter 10, "Reconfiguring the Application Server"](#)

What is in this Book?

This book contains:

- Information about managing an application server as a single entity. It describes how to design and install an application server, start and stop an application server, and manage users, security, availability, and clustering.
- An introduction to managing the individual components that make up an application server. It provides getting started information for managing components using a centralized tool, and illustrates how to use the tool by describing common configuration, deployment, and monitoring tasks for several key components. It also provides information about dependencies between components.

Related Administration Documentation

In addition to this book, Oracle offers many other books with information about managing Oracle9iAS. These books are listed in [Table 1–1](#). All books are available in the Oracle9iAS Documentation Library, except where noted.

Table 1–1 Related Administration Documentation

For information about:	See Also:
Getting Started	
Oracle9iAS architecture, solutions, and features	<i>Oracle9i Application Server Concepts Guide</i>
Oracle9iAS performance monitoring and tuning	<i>Oracle9i Application Server Performance Guide</i>
Oracle9iAS security	<i>Oracle9i Application Server Security Guide</i>
Configuring Oracle9iAS for global application deployment	<i>Oracle9i Application Server Globalization Support Guide</i>
Managing Components	
mod_plsql	<i>mod_plsql User's Guide</i>
Oracle HTTP Server	<i>Oracle HTTP Server Administration Guide</i>
Oracle Internet Directory	<i>Oracle Internet Directory Administrator's Guide</i>
Oracle9iAS Clickstream Intelligence	<i>Oracle9iAS Clickstream Intelligence Administrator's Guide</i>
Oracle9iAS Containers for J2EE	<i>Oracle9iAS Containers for J2EE User's Guide</i>
Oracle9iAS Discoverer	<i>Oracle9iAS Discoverer Configuration Guide</i>
Oracle9iAS Forms Services	<i>Oracle9iAS Forms Services Deployment Guide</i>
Oracle9iAS Personalization	<i>Oracle9iAS Personalization Administrator's Guide</i>
Oracle9iAS Portal	<i>Oracle9iAS Portal Configuration Guide</i>
Oracle9iAS Reports Services	<i>Oracle9iAS Reports Services Publishing Reports to the Web</i>
Oracle9iAS Single Sign-On	<i>Oracle9iAS Single Sign-On Administrator's Guide</i>
Oracle9iAS Syndication Server	<i>Oracle9iAS Syndication Server User's and Administrator's Guide</i> (available in the Oracle9i Developer Suite documentation library)
Oracle9iAS Unified Messaging	<i>Oracle9iAS Unified Messaging Administrator's Guide</i>
Oracle9iAS Web Cache	<i>Oracle9iAS Web Cache Administration and Deployment Guide</i>
Oracle9iAS Wireless	<i>Oracle9iAS Wireless Getting Started and System Guide</i>

Table 1–1 Related Administration Documentation (Cont.)

For information about:	See Also:
Management Tools	
Oracle Enterprise Manager architecture and features	<i>Oracle Enterprise Manager Concepts Guide</i>
Configuring Oracle Enterprise Manager	<i>Oracle Enterprise Manager Configuration Guide</i>
Managing Oracle Enterprise Manager	<i>Oracle Enterprise Manager Administrator's Guide</i>
Managing Databases	
Managing the Oracle9 <i>i</i> databases in your enterprise	<i>Oracle9<i>i</i> Administrator's Guide</i> (available in the Oracle9 <i>i</i> Database documentation library)

2

Concepts for Administrators

This chapter introduces concepts for designing, installing, and managing an application server.

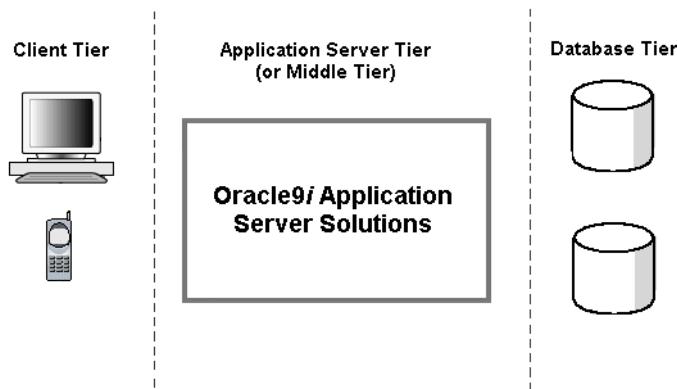
It contains the following topics:

- [Introduction](#)
- [Application Server Installations](#)
- [The Oracle9iAS Infrastructure](#)
- [Customer Databases](#)
- [Sample Application Server Enterprises](#)
- [Designing an Application Server Enterprise](#)
- [Installing an Application Server Enterprise](#)
- [Managing an Application Server Enterprise](#)
- [Reconfiguring an Application Server Enterprise](#)

Introduction

Oracle9*i* Application Server uses a three-tiered architecture as shown in [Figure 2–1](#).

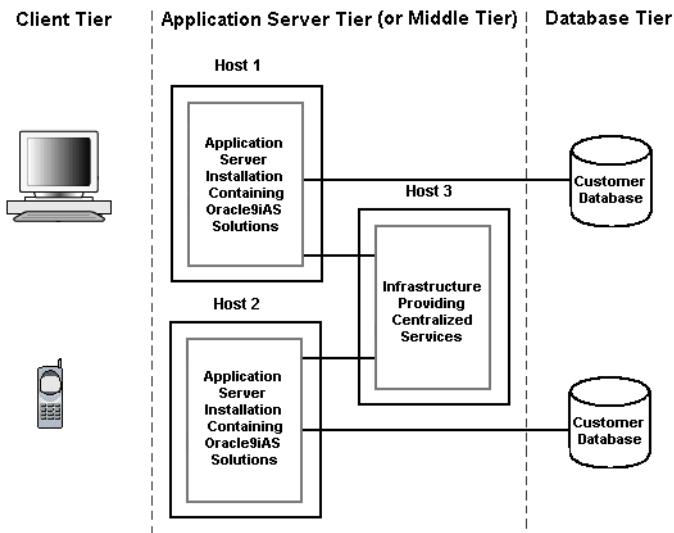
Figure 2–1 Oracle9*i*AS Three-Tiered Architecture



- The *client tier* contains different types of clients including Web browsers, mobile and wireless devices, Web Services, and back-office applications. Clients send requests to the application server tier using a variety of physical networks and protocols.
- The *application server tier*, or *middle tier*, contains the Oracle9*i*AS solutions that provide industry-standard Web applications and services. The application server tier processes requests from the client tier. It may access or update data in the database tier to complete a request.
- The *database tier* consists of databases that contain data, metadata, and business logic.

To clients, the application server tier and database tier appear to be a single, ever-present entity that efficiently processes requests. In reality, the application server tier and database tier may contain many entities working together to create a seamless platform, as shown in [Figure 2–2](#). There may be one installation of Oracle9*i*AS solutions, called an *application server installation*, on a single host, or multiple application server installations on multiple hosts. There may be a central source of user, security, and configuration information called an *infrastructure*. There may be databases in the database tier, called *customer databases*, that collect and process information for Oracle9*i*AS solutions.

Figure 2–2 Application Server Enterprise in the Three-Tiered Architecture



This collection of application server installations, infrastructures, and customer databases is called an *application server enterprise*. The application server enterprise is the framework within which you design, install, and manage all of the features and solutions Oracle9i Application Server offers.

This chapter introduces the concepts and practices for managing an application server enterprise.

Application Server Installations

An application server installation (also called a *middle-tier application server installation* or *middle-tier installation*) is a set of Oracle9iAS solutions installed into a single Oracle home. The Oracle9iAS solutions in an application server installation are called *components*. Your enterprise can have one or more application server installations, residing on one host or distributed across multiple hosts.

Application Server Install Types

There are four types of application server installations, or *install types*, each providing a set of components. The install types are ordered, in that each contains

all of the components in the previous install type, plus additional components. The install types are:

- **J2EE and Web Cache:** Provides a basic Web server that enables you to develop and deploy Java 2 Enterprise Edition (J2EE) applications, use J2EE and Simple Object Access Protocol (SOAP) based Web services, accelerate Web site performance, and manage your application server with a Web-based tool.
- **Portal and Wireless:** Enables you to deploy enterprise portals and wireless applications. Includes all components available in the J2EE and Web Cache install type.
- **Business Intelligence and Forms:** Enables you to analyze clickstream data, personalize applications, perform data analysis, generate Web-based reports, create Discoverer public connections, configure Discoverer viewer and Discoverer Portlet Provider, and use Forms-based applications. Includes all components available in the Portal and Wireless install type.
- **Unified Messaging:** Enables you to integrate different types of messages into a single framework. Includes all of the components available in the Business Intelligence and Forms install type.

[Table 2-1, "Oracle9iAS Install Types"](#) shows which components are provided in each install type.

Table 2–1 Oracle9iAS Install Types

Component	J2EE and Web Cache	Portal and Wireless	Business Intelligence and Forms	Unified Messaging
Oracle HTTP Server ¹	YES	YES	YES	YES
Oracle9iAS Containers for J2EE	YES	YES	YES	YES
Oracle9iAS Web Cache	YES	YES	YES	YES
Oracle Enterprise Manager Web site	YES	YES	YES	YES
Oracle9iAS Portal ²	no	YES	YES	YES
Oracle9iAS Wireless	no	YES	YES	YES
Oracle9iAS Discoverer ³	no	no	YES	YES
Oracle9iAS Reports Services	no	no	YES	YES
Oracle9iAS Clickstream Intelligence	no	no	YES	YES
Oracle9iAS Forms Services	no	no	YES	YES
Oracle9iAS Personalization	no	no	YES	YES
Oracle9iAS Unified Messaging	no	no	no	YES

¹ Oracle HTTP Server installs the following Oracle mods: mod_oc4j, mod_oss, mod_proxy, DMS, FastCGI, HiAv Infrastructure, Oracle9iAS Java Object Caching, SOAP, DBI/DBD, Clickstream Collector Agent.

² Oracle9iAS Portal installs Oracle Ultra Search and Oracle9i Syndication Server.

³ Oracle9iAS Discoverer installs Discoverer Viewer, Discoverer Plus, and Discoverer Portlet Provider.

See Also: [Appendix A, "Component Descriptions"](#) for more information on each component

About Configuring Components

An application server installation contains the files necessary to support all of its components, but you do not have to configure all components.

Note: The term *configure* can have one of two meanings, depending on the status of a component. The first time a component is configured, it is initially setup to run. After that, to configure a component means to tune component parameters. This chapter uses the term *configure* to mean the initial setup of a component.

The following components are automatically configured:

- Oracle HTTP Server (automatically configured in all install types)
- Oracle9iAS Containers for J2EE (automatically configured in all install types)
- Oracle9iAS Web Cache (automatically configured in Portal and Wireless, Business Intelligence and Forms, and Unified Messaging install types; optional in the J2EE and Web Cache install type)

Other components can be configured at install time or any time after installation. Because some components depend on each other, there are a few guidelines for configuring them, as described in [Table 2–2](#).

Table 2–2 Guidelines for Configuring Components

If you configure:	You should configure:
Oracle9iAS Clickstream Intelligence	Oracle9iAS Discoverer Oracle9iAS Portal
Oracle9iAS Discoverer	Oracle9iAS Portal
Oracle9iAS Forms Services	Oracle Management Server <p><i>Note: Oracle Management Server is part of an infrastructure installation. See "The Oracle9iAS Infrastructure" on page 2-7 for more information.</i></p>
Oracle9iAS Reports Services	Oracle9iAS Portal
Oracle9iAS Wireless	Oracle9iAS Portal

Once you configure a component, it cannot be deconfigured, but you can remove its functionality by not starting it when you start the rest of the components in your installation.

The Oracle9iAS Infrastructure

An Oracle9iAS Infrastructure is a type of installation that provides centralized security and management services, configuration information, and data repositories for application server installations.

An infrastructure is installed into its own Oracle home and contains the following pieces:

- **Oracle9iAS Single Sign-On:** This feature enables users to access multiple accounts and applications with a single username and password, and provides single sign-on service across your entire application server enterprise.
- **Oracle Internet Directory:** This is an LDAP-compliant directory service that provides centralized storage of information about users, applications, and resources in your enterprise.
- **Metadata Repository:** This is an Oracle9i Enterprise Edition database that contains schemas and business logic used by application server components and other pieces of the infrastructure.
- **Oracle Management Server (optional):** This is required if you would like to use Oracle Enterprise Manager Console, a Java-based interface for managing the application server instances, databases, and applications in your enterprise.

See Also: ["About the Oracle Enterprise Manager Console"](#) on page 3-18 for more information

When is an Infrastructure Required?

An infrastructure is required for most install types. The one exception is a J2EE and Web Cache installation that is not intended to use Single Sign-On or clustering. Clustering is an operation that involves joining application server installations together to share configuration parameters and application deployment.

See Also: [Chapter 14, "Application Server Clustering"](#)

Table 2-3 summarizes the infrastructure requirements for application server install types. The infrastructure must be installed before any middle-tier application server installations that are going to use it.

Table 2–3 Infrastructure Requirements for Application Server Install Types

Application Server Install Type	Infrastructure Required?
J2EE and Web Cache (not using Oracle9iAS Single Sign-On or clustering)	No
J2EE and Web Cache (using Oracle9iAS Single Sign-On or clustering)	Yes
Portal and Wireless	Yes
Business Intelligence and Forms	Yes
Unified Messaging	Yes

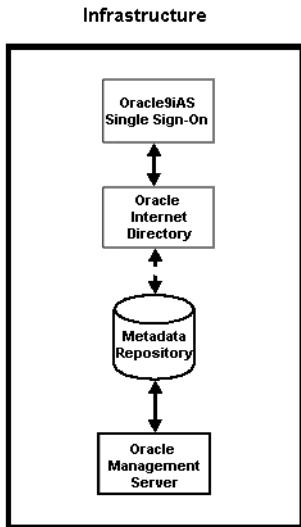
How Infrastructure Pieces Work Together

The pieces of an infrastructure work together as follows:

- Oracle9iAS Single Sign-On stores user information in Oracle Internet Directory.
- Oracle Internet Directory uses the metadata repository as its back-end database.
- Oracle Management Server uses the metadata repository to store information

[Figure 2–3](#) shows the relationship between the pieces of an infrastructure. Note that all pieces of the infrastructure are installed into a single Oracle home.

Figure 2–3 Relationship Between Pieces of an Infrastructure



How Application Server Installations Use An Infrastructure

Application server installations and their components use an infrastructure in the following ways:

- Components and applications use the Single Sign-on service provided by Oracle9iAS Single Sign-On.
- Application server installations and components store configuration information and user and group privileges in Oracle Internet Directory.
- Components use schemas that reside in the metadata repository.

Configuring an Application Server Installation to Use an Infrastructure

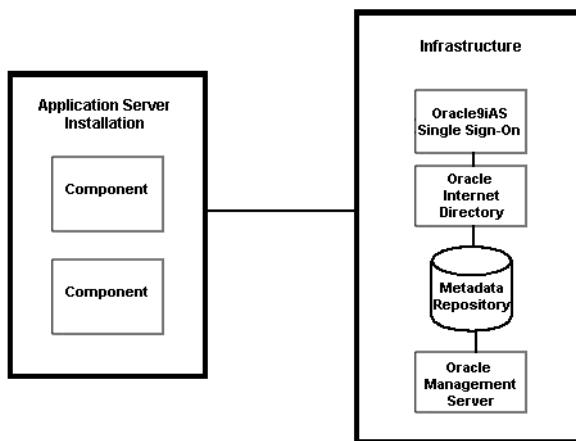
You can configure an application server installation to use an infrastructure at the following times:

- During the installation of any application server install type. The Oracle Universal Installer guides you through this and it is required for all install types except J2EE and Web Cache.

- After the installation of a J2EE and Web Cache install type. You can associate a J2EE and Web Cache install type with an infrastructure at any time if you want to use Single Sign-On or clustering.

Figure 2–1 shows the convention used throughout this book to depict an application server installation using an infrastructure. Note that the application server installation and the infrastructure are each installed into their own Oracle home and may reside on a single host or on separate hosts.

Figure 2–4 Application Server Installation Using an Infrastructure



Customer Databases

Most Oracle9iAS components use schemas in the metadata repository. However, several components require their schemas to be co-resident with customer data in a specially-tuned database called a *customer database*. Customer databases are Oracle databases that reside on any system in your enterprise.

Components that Require a Customer Database

The following components require a customer database:

- Oracle9iAS Discoverer
- Oracle9iAS Personalization
- Oracle9iAS Unified Messaging

If you configure any of these components during installation, their setup and configuration will not be complete at the end of installation. You need to take additional steps to install and tune a customer database, load schemas into the database, and finish configuring the component to use the customer database.

If you configure any of these components after installation, you will take the same steps to install and tune a customer database, load schemas into the database, and configure the component to use the customer database.

Components that Can Use a Metadata Repository or Customer Database

The following components can use a metadata repository or a customer database:

- Oracle9iAS Clickstream Intelligence
- Oracle9iAS Portal

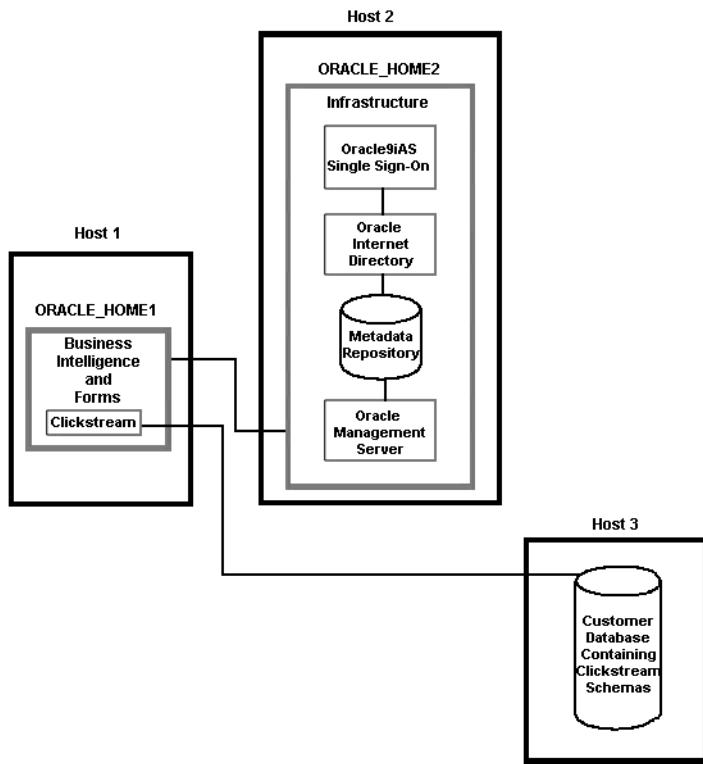
If you configure either of these components during installation, they are automatically configured to use the metadata repository and are completely configured and running at the end of installation.

If you configure either of these components after installation, you can use the metadata repository or a customer database. If you use a customer database, you must install and tune the database and load schemas into it as part of the configuration.

Depiction of a Component Using a Customer Database

When you configure an individual component to use a customer database, it does not affect the rest of the components in the installation; they still use the metadata repository. [Figure 2–5](#) shows the convention used throughout this book to depict a component using a customer database. In this figure, all components within the application server installation on Host 1 use the metadata repository on Host 2, except Clickstream, which uses a customer database on Host 3.

Figure 2–5 A Component Using a Customer Database



Sample Application Server Enterprises

This section shows sample application server enterprises that illustrate the configuration options discussed in this chapter.

Figure 2–6 depicts the simplest application server enterprise: a single J2EE and Web Cache installation with no infrastructure. This installation cannot use Single Sign-On or clustering; however, it is a fully functional J2EE platform with caching capabilities.

Figure 2–6 Single J2EE and Web Cache Installation

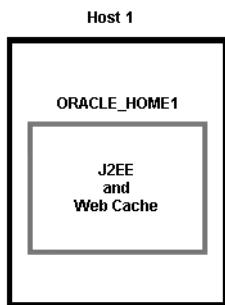


Figure 2–7 depicts two application server installations on one host. In this design, both install types are J2EE and Web Cache that do not use Single Sign-On or clustering, so no infrastructure is required. Note that you can mix different install types on the same host. For example, you can install J2EE and Web Cache and Portal and Wireless on the same host.

Figure 2–7 Two J2EE and Web Cache Installations

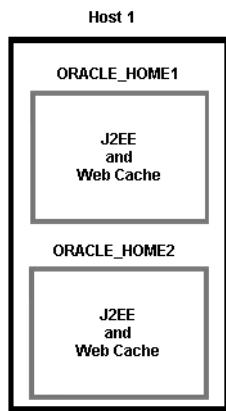


Figure 2–8 shows two J2EE and Web Cache installations on a single host and both use an infrastructure on another host. These application server installations can use Single Sign-On services and clustering because they are associated with an infrastructure. Note that the infrastructure could be on the same host as the J2EE and Web Cache installations, with all three installations using a different Oracle home.

Figure 2–8 Two J2EE and Web Cache Installations with an Infrastructure

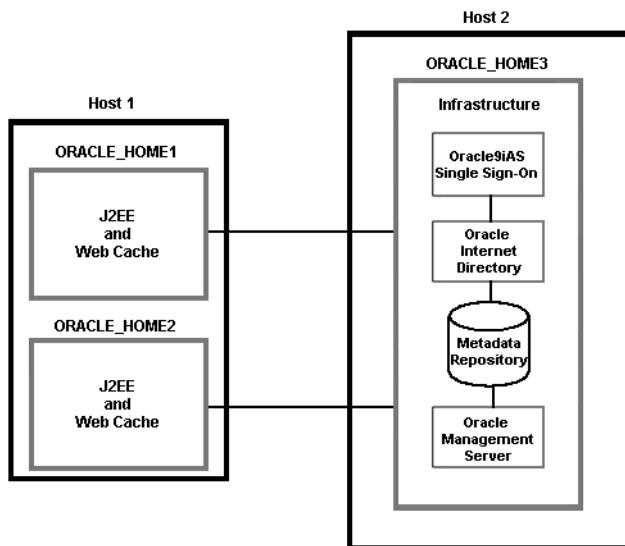


Figure 2–9 shows that two different install types can exist on the same host. Host 1 contains a J2EE and Web Cache installation and a Portal and Wireless installation. It also shows that installations on multiple hosts can use the same infrastructure. The installations on Host1 and Host 3 all use the same infrastructure on Host 2.

Figure 2–9 Installations on Multiple Hosts Using the Same Infrastructure

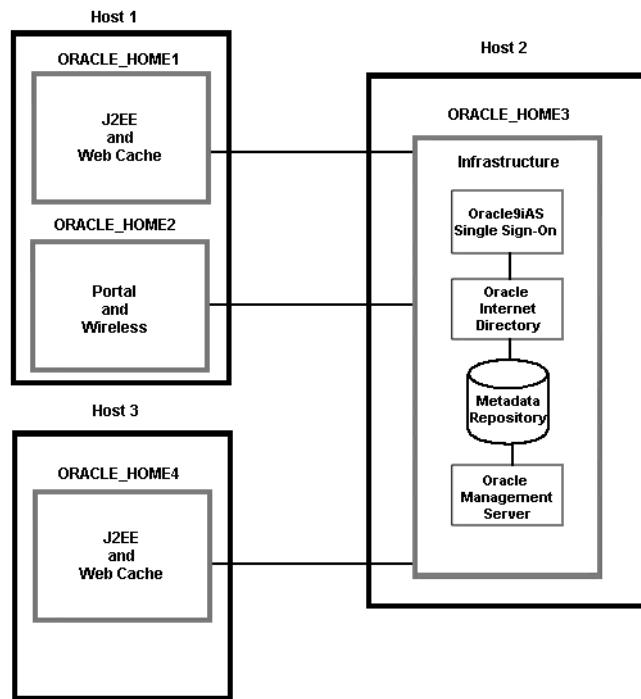
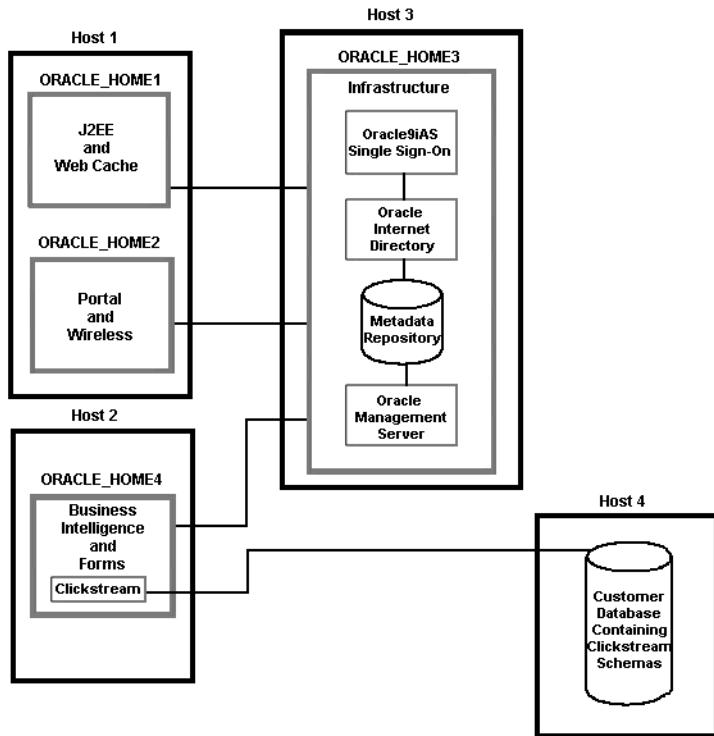


Figure 2–10 depicts a single component using a customer database. In this design, all components in the application server installations on Host 1 and Host 3 use the metadata repository on Host 3, except the Clickstream component on Host 2. Clickstream uses the customer database on Host 4.

Figure 2–10 Component Using a Customer Database



Designing an Application Server Enterprise

This section describes the following steps for designing an application server enterprise:

- Step 1: Choose Your Application Server Install Types
- Step 2: Choose Which Components to Configure
- Step 3: Determine Your Infrastructure Requirements
- Step 4: Choose Your Hosts
- Step 5: Determine Your Customer Database Requirements
- Step 6: Draw Your Application Enterprise Design

Step 1: Choose Your Application Server Install Types

Determine which of the following application server install types you would like to install:

- J2EE and Web Cache
- Portal and Wireless
- Business Intelligence and Forms
- Unified Messaging

Remember, you can start with a single installation and add more installations later.

Step 2: Choose Which Components to Configure

Choose the components you would like to configure. You do not need to configure all components at install time because you can configure components after installation. Also, if you want to use Oracle9iAS Clickstream Intelligence or Oracle9iAS Portal with a customer database, do not configure them during installation; configure them after installation.

Step 3: Determine Your Infrastructure Requirements

If you intend to use any of the following install types, you require an infrastructure:

- J2EE and Web Cache (using Oracle9iAS Single Sign-On or clusters)
- Portal and Wireless
- Business Intelligence and Forms
- Unified Messaging

You must configure Oracle9iAS Single Sign-On, Oracle Internet Directory, and the metadata repository in your infrastructure. If you intend to use Oracle Enterprise Manager Console, Oracle9iAS Forms Services, or Oracle9iAS Interconnect you should also configure Oracle Management Server.

See Also: It is possible to leverage from an existing Oracle Internet Directory in your enterprise. Refer to *Oracle Internet Directory Administrator's Guide* before proceeding with these steps.

Step 4: Choose Your Hosts

Choose the hosts you will use for application server installations and, optionally, an infrastructure. The following guidelines apply to choosing a host:

- Make sure each host satisfies the operating system and hardware requirements for the installations you intend to put on it. It is recommended that you install an infrastructure on a different host than your application server installations, but you can install an infrastructure and an application server on the same host if the host meets the minimum hardware requirements.

See Also: *Oracle9i Application Server Installation Guide* and platform-specific installation guides for more information

- The hosts you use must have a fixed IP address and hostname. They cannot use a dynamic IP address management system such as DHCP.
- Consider whether you want to distribute application server installations over multiple hosts to implement high-availability schemes or clustering.

See Also: [Chapter 13, "High Availability"](#) and [Chapter 14, "Application Server Clustering"](#) for more information

Step 5: Determine Your Customer Database Requirements

If you intend to configure any of the following components, you must provide a customer database:

- Oracle9iAS Discoverer
- Oracle9iAS Personalization
- Oracle9iAS Unified Messaging

If you intend to configure the following components, you can configure them to use a metadata repository during installation or you can configure them to use a customer database after installation:

- Oracle9iAS Clickstream Intelligence (it is recommended that you use a customer database for Clickstream if you intend to acquire and store a large amount of data)
- Oracle9iAS Portal

Customer databases can be databases that already exist in your data center, or can be newly installed for use with the application server. In either case, you must make sure that they allow remote access by SYS so components can load data into them. They must be the appropriate version and be properly tuned for the component they support.

See Also: Some components require an Oracle9*i* database and some will work with an Oracle8*i* database. See component administration guides for information on customer database requirements.

Step 6: Draw Your Application Enterprise Design

After completing the preceding steps, you should be able to draw an application server enterprise design similar to the ones shown in this chapter. You are now ready to install your application server enterprise.

Installing an Application Server Enterprise

The Oracle Universal Installer guides you through installing and configuring your application server enterprise. This section discusses rules and concepts you should be familiar with before installing an application server, including:

- [Oracle Homes](#)
- [Coexistence With Other Products](#)
- [Port Allocation](#)
- [Primary and Secondary Installations](#)
- [About Infrastructure Association](#)

See Also: *Oracle9i Application Server Installation Guide* for complete installation instructions

Oracle Homes

The following rules apply to Oracle homes used by application server installations and infrastructures:

- You must install each application server installation in its own Oracle home. No other products can be installed in that Oracle home, except Oracle9i Developer Suite (Oracle9iDS).
- You must install an infrastructure in its own Oracle home. No other products (including Oracle9i Developer Suite) can be installed in that Oracle home.
- You can install multiple application server installations and infrastructures on the same host. Each must have its own Oracle home.
- You must use the same operation system user account when installing application server installations and infrastructure on the same host. For example, if you log in as the user "oracle" to perform the first installation on a host, you must log in as "oracle" to perform all subsequent installations on that host.

Coexistence With Other Products

This section describes how application server installations and infrastructures can coexist with other products on the same system:

- You can install application server installations and infrastructures on a host that has other Oracle databases and pre-9.0.2 versions of Oracle application servers.

Each product must have its own Oracle home and use different ports. Note that the host must have sufficient hardware to support multiple installations.

- The metadata repository is always configured to use port 1521 and has a service name of the form `iasdb.host.domain` (for the first infrastructure on a host) and `iasdbx.host.domain` for any subsequent infrastructures on that host). You should never change these values. If you have another database on the host using the same service name, you must change it before installing the metadata repository.

If you have another application on the host using port 1521, you must do one of the following *before* installing the infrastructure:

- If the other application is an Oracle Net listener using an Oracle9*i* database, no action is required. The metadata repository will use the existing Oracle Net listener.
- If the other application is an Oracle Net8 listener using an Oracle8*i* database, you must change the Net8 listener to use a different port than 1521, or you must upgrade it to the Oracle9*i* Net listener that runs with the metadata repository.

See Also: *Oracle9i Application Server Installation Guide* for instructions on upgrading an existing Net8 listener

- If the other application is not an Oracle Net or Oracle Net8 listener, reconfigure it to use a different port.

Port Allocation

The application server automatically assigns port numbers for many services during installation. It chooses port numbers from a set of default port numbers and port ranges.

See Also: [Appendix D, "Default Port Numbers and Port Ranges"](#)

During installation, the application server assigns unique port numbers in the following ways:

- It internally tracks the port numbers used by all application server installations and infrastructures (running or not running) on a host and will not use the same port number twice.

- It performs a system check to determine if any *non*-Oracle9iAS processes are running on the port.

Despite this strategy, a port conflict may still occur between the application server and a *non*-Oracle9iAS process that was not running during the application server installation. Oracle recommends that you review the application server port numbers assigned during installation to determine if there are any port conflicts with the rest of your system.

See Also: ["Task 2: Check for Port Conflicts" on page 4-3](#)

Primary and Secondary Installations

If you install multiple application server installations and infrastructures on one host, the first installation is called the *primary installation*. All succeeding installations on that host are called *secondary installations*.

Primary installations are unique because they contain the configuration information and password for Enterprise Manager Web site, the Oracle9iAS management tool. When you install a primary installation, you are asked to set the `ias_admin` password. The `ias_admin` user is required to use the Enterprise Manager Web site. The `ias_admin` password is the same for all installations on one host and you must supply it when you install any secondary installations on a host.

See Also: ["About the Oracle Enterprise Manager Web Site" on page 3-2](#) for more information

About Infrastructure Association

Once a middle-tier installation is associated with an infrastructure, all subsequent middle-tier installations on the same host will be associated with that infrastructure. This section describes some possible scenarios:

- Scenario 1:
 1. Install an infrastructure on Host A.
 2. Install Portal and Wireless on Host B and associate it with the infrastructure on Host A.
 3. Install J2EE and Web Cache on Host B. If you choose to use an infrastructure, it will automatically use the infrastructure on Host B.
- Scenario 2:
 1. Install an infrastructure on Host A.

2. Install Portal and Wireless on Host B and associate it with the infrastructure on Host A.
 3. Install Business Intelligence and Forms on Host B. It will automatically use the infrastructure on Host A.
- Scenario 3:
 1. Install an infrastructure on Host A.
 2. Install J2EE and Web Cache on Host A and choose to use an infrastructure. You can choose the infrastructure on Host A or any other infrastructure.
 3. Install Portal and Wireless on Host A. It will automatically use the same infrastructure that the J2EE and Web Cache installation uses.
 - Scenario 4:
 1. Install an infrastructure on Host A.
 2. Install a J2EE and Web Cache on Host B. Do not choose to use an infrastructure.
 3. Install Portal and Wireless on Host B and associate it with the infrastructure on Host A. The J2EE and Web Cache installation will continue to not use an infrastructure. Any subsequent middle-tier installations will automatically use the infrastructure on Host A.

Managing an Application Server Enterprise

Once you have installed your application server, you can start managing it. This section introduces the entities that play a primary role in managing the application server. These entities are discussed throughout this book and are as follows:

- [Application Server Instances](#)
- [Component Instances](#)
- [Farms](#)
- [Clusters](#)

Application Server Instances

An *application server instance* is the set of processes required to run the configured components within an application server installation. There can only be one application server instance per application server installation. The terms installation

and instance are sometimes used interchangeably; however, it is important to remember that an installation is the set of files installed into an Oracle home and an instance is the set of processes associated with those files.

All application server instances have a name, which you set during installation. You can use this name to identify the instance when using the Oracle9iAS management tools. The hostname is appended to the name you supply to form the instance name. For example, if you assign the name `ias1` to an instance on host `myserver.mycompany.com`, the instance name is `ias1.myserver.mycompany.com`.

Component Instances

A *component instance* is the set of processes required to run a component within an application server instance. When you configure a component, either during installation or after installation, one or more component instances are created that you can use to start, stop, and manage the component. This section contains the following two tables that describe component instances:

- [Table 2-4, "Component Instances in Application Server Installations"](#)
- [Table 2-5, "Component Instances in Infrastructures"](#)

Table 2-4 Component Instances in Application Server Installations

Install Type	Component Instance	Description
J2EE and Web Cache	BC4J	Business Components for Java. This is a framework for building applications and is deployed to the OC4J_home instance.
	Clickstream Collector	Clickstream Collector Agent. Although Clickstream is a part of the Business Intelligence and Forms install type, the Clickstream Collector Agent is included with every install type. It collects data from the HTTP Server for Clickstream analysis. You can only use the Clickstream Collector if you are using Clickstream.
	HTTP Server	Oracle HTTP Server instance.
	JServ	The Java servlet environment in HTTP Server. This component can only be configured after installation.
	OC4J_Demos	An OC4J instance that contains OC4J demos.
	OC4J_home	The default OC4J instance that is created when OC4J is configured. You can use this to deploy applications.
	Additional OC4J Instances	You can create additional OC4J instances after installation and assign them a name.
	Web Cache	Oracle9iAS Web Cache instance.

Table 2–4 Component Instances in Application Server Installations (Cont.)

Install Type	Component Instance	Description
Portal and Wireless (includes all component instances in J2EE and Web Cache)	OC4J_Portal	An OC4J instance that contains a servlet to support Portal.
	OC4J_Wireless	An OC4J instance that contains a servlet to support Wireless.
	Portal	Oracle9iAS Portal instance.
	Single Sign-On	Oracle9iAS Single Sign-On instance.
Business Intelligence and Forms (includes all component instances in Portal and Wireless)	Wireless	Oracle9iAS Wireless instance.
	Clickstream	Oracle9iAS Clickstream Intelligence instance.
	Discoverer	Oracle9iAS Discoverer instance.
	Forms	Oracle9iAS Forms Services instance.
	OC4J_BI_Forms	An OC4J instance that runs servlets that support Discoverer, Clickstream, Forms, Personalization, and Reports.
	Personalization	Oracle9iAS Personalization instance.
	Reports	Oracle9iAS Reports Services instance.
Unified Messaging (includes all component instances in Business Intelligence and Forms)	Unified Messaging	Oracle9iAS Unified Messaging instance.

Table 2–5 Component Instances in Infrastructures

Install Type	Component Instance	Description
Infrastructure	BC4J	Business Components for Java. This instance is not used by the infrastructure.
	Clickstream Collector	Clickstream Collector Agent. This instance is not used by the infrastructure.
	HTTP Server	Oracle HTTP Server instance.
	Internet Directory	Oracle Internet Directory
	OC4J_DAS	An OC4J instance that provides the Delegated Administration Service for Oracle Internet Directory.
	OC4J_Demos	An OC4J instance that contains OC4J demos. This instance is not used by the infrastructure.
	OC4J_home	The default OC4J instance. This instance is not used by the infrastructure.
	Single Sign-On	Oracle9iAS Single Sign-On

Farms

A *farm* is a set of one or more application server instances that use the same metadata repository. Exactly one farm is associated with a metadata repository and it has a name of the form *host.port.sid*, where *host* is the fully qualified name of the host on which the metadata repository resides, *port* is the metadata repository listener port, and *sid* is the metadata repository service id. An example of a farm name is `myserver.mycompany.com.1521.iasdb`.

The main purpose of a farm is that it designates a set of application server instances that share a repository of configuration and deployment information within the same metadata repository, and are therefore eligible to be clustered together. Clustered instances must share configuration and application information, and membership in a farm enables instances to share information.

Clusters

A *cluster* is a collection of application server instances with identical configuration parameters and application deployment. Clusters enforce homogeneity between member instances so that a cluster of application server instances appears and functions as a single instance. With appropriate front-end load balancing, any instance in a cluster can serve client requests. Only J2EE and Web Cache install types can be clustered.

See Also: [Chapter 14, "Application Server Clustering"](#) for more information

Reconfiguring an Application Server Enterprise

Once your application server is up and running, you may find you need to reconfigure it to provide more functionality or increase its capacity. You can reconfigure your application server in the following ways:

- **Configure Components:** You can add functionality by configuring components that were not configured at install time.
- **Extend application server installations:** You can extend an application server installation by installing a higher install type in the same Oracle home. For example, you can extend a Portal and Wireless installation to a Business Intelligence and Forms installation or a Unified Messaging installation.
- **Add an infrastructure:** You can add your first infrastructure to your application server enterprise in order to enable Single Sign-On or clustering for existing J2EE and Web Cache installations, or you can add a second infrastructure if the load on the metadata repository is too great.
- **Associate an instance with an infrastructure (join a farm):** You can associate a J2EE and Web installation with an infrastructure, thereby adding it to a farm.
- **Change the Database Used By a Component:** You can change the database used by an individual component. This is usually done for migration purposes.

See Also: [Chapter 10, "Reconfiguring the Application Server"](#) for details on how to perform the operations described in this section

3

Introduction to Administration Tools

This chapter introduces the Oracle9*i* Application Server administration tools.

It contains the following topics:

- [Introduction](#)
- [About the Oracle Enterprise Manager Web Site](#)
- [About the Oracle Enterprise Manager Console](#)
- [Using the Console with the Oracle Enterprise Manager Web Site](#)

Introduction

This chapter introduces the tools you can use to manage your Oracle9iAS environment. The primary tool for managing Oracle9iAS—as well as your entire Oracle environment—is Oracle Enterprise Manager.

Many Oracle users are familiar with the Oracle Enterprise Manager Console, which is designed to reduce the complexity of managing your Oracle Database Server as well as the network components, software, and hardware that support your Oracle environment.

With Oracle9iAS Release 2, Oracle introduces the Oracle Enterprise Manager Web site, which is designed specifically to help you manage Oracle9i Application Server.

Together, the Oracle Enterprise Manager Console and the Oracle Enterprise Manager Web site provide a complete set of efficient tools to reduce the cost and complexity of managing your enterprise.

About the Oracle Enterprise Manager Web Site

The Oracle Enterprise Manager Web site provides Web-based management tools designed specifically for Oracle9iAS. Using the Web site, you can monitor and configure the components of your Oracle9i Application Server installations. You can deploy applications, manage security, and create and manage Oracle9i Application Server clusters.

The Oracle Enterprise Manager Web site consists of:

- The Enterprise Manager Home Pages you use to manage Oracle9iAS
 - These Web pages provide you with a high-level view of your Oracle9i Application Server environment. For more information, see "[Introducing the Enterprise Manager Home Pages](#)" on page 3-3.
- The underlying software technologies that keep track of your application server instances and components
 - These technologies automatically perform many management tasks. For example, they discover the components of each application server instance, gather and process performance data, and provide access to application configuration information. For more information, see "[About the Underlying Technologies](#)" on page 3-17.

Introducing the Enterprise Manager Home Pages

Oracle9iAS provides a wide variety of software solutions designed to help you run all aspects of your business. As a result, you will want to manage Oracle9iAS from different levels of detail. At times, you may want to manage a set of application server instances to compare their performance; or, you may find it efficient to combine multiple instances into an Oracle9iAS cluster.

At other times, you will want to manage an individual application server instance or a specific component--or even a specific Web application used to run your business or support your customers.

Further, different administrators may specialize in managing specific aspects of your Oracle9iAS environment. One administrator may monitor your J2EE application deployments. Another may be responsible for managing your implementation of Oracle9iAS Portal.

To support these multiple levels of management, Oracle introduces the Oracle Enterprise Manager Home Pages. Each Home Page provides the information you need to monitor the performance and availability of Oracle9iAS from a particular level of management detail. Selected Home Pages also provide tools for configuring your Oracle9iAS components.

From each Home Page, you can obtain high-level information or you can drill down to get more specific information about an instance, component, or application.

Consider the following Home Pages that are available when you use the Oracle Enterprise Manager:

- Use the Oracle9iAS Farm Home Page to view a set of related application server instances on your network and to create clusters that speed up the configuration and deployment of your Web applications. For more information, see "[Using the Oracle9iAS Farm Home Page](#)" on page 3-7.
- Use the Oracle9iAS Instance Home Page to manage all aspects of an individual application server instance. For more information, see "[Using the Oracle9iAS Instance Home Page](#)" on page 3-5.
- Drill down to a component Home Page to monitor or configure an individual component of the application server. For example, use the Oracle HTTP Server Home Page to monitor the performance of your Web server, or use the Oracle Containers for J2EE (OC4J) Home Page to deploy a custom Web-based application. For more information, see "[Using an Oracle9iAS Component Home Page](#)" on page 3-8.

Getting Started with the Oracle Enterprise Manager Web Site

To view the Oracle Enterprise Manager Web site, click **Log on to Oracle Enterprise Manager** on the Oracle9iAS Welcome page.

You can access the Welcome Page by pointing your browser to the HTTP Server URL for your installation. For example, the default HTTP Server URL is:

`http://hostname:7777`

You can also go directly to the Oracle Enterprise Manager Web site using the following instructions:

1. Enter the following address in your Web browser:

`http://hostnanme:<port_number>/`

For example, if you installed Oracle9i Application Server on a host called SYS12, you would enter the following address in your Web browser:

`http://SYS12:1810`

Note: Experienced administrators can modify the default port, but in most cases, the port specified for the Oracle Enterprise Manager Web site is 1810.

2. Enter the Oracle9i Application Server administrator username and password.

The username for the administrator user is `ias_admin`. The password is defined during the installation of Oracle9iAS.

3. Click **OK**.

Depending upon the components you configured and the choices you made during the Oracle9iAS installation, Enterprise Manager displays the appropriate Home Page.

The initial Home Page varies depending upon whether or not you installed an Oracle9iAS Infrastructure. For more information about the options available when installing Oracle9iAS, see the *Oracle9i Application Server Installation Guide*.

Table 3-1 describes the Enterprise Manager Home Pages that might be used as a starting point when you first browse to the Oracle Enterprise Manager Web site.

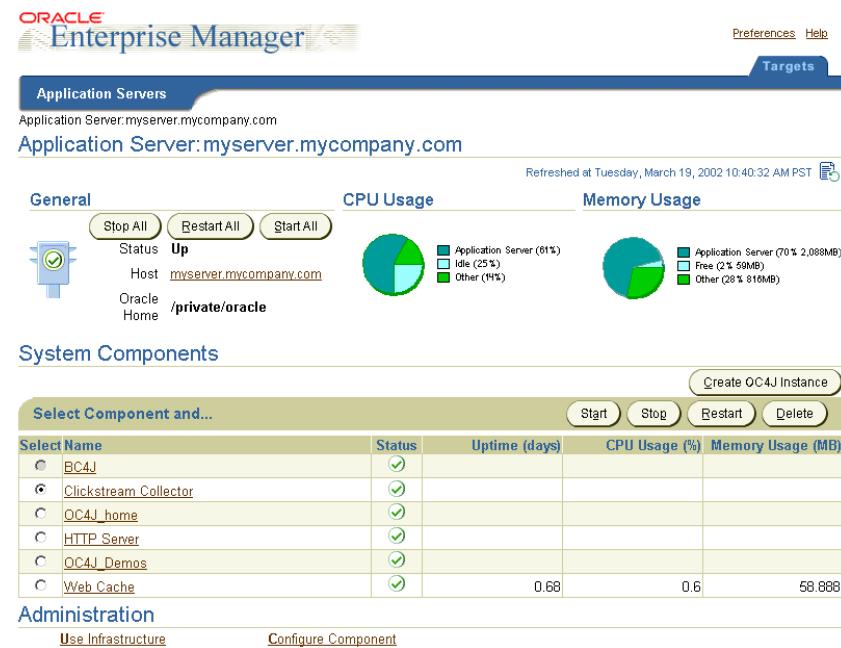
Table 3–1 Enterprise Manager Home Pages for Managing Oracle9iAS

Enterprise Manager Home Page	Description
Oracle9iAS Instance Home Page	<p>Use this Home Page to monitor and configure a single application server instance.</p> <p>For more information, see "Using the Oracle9iAS Instance Home Page" on page 3-5.</p>
Oracle9iAS Farm Home Page	<p>The Instance Home Page is the first page you see if you have installed a single application server instance that is not associated with an Oracle9iAS Infrastructure.</p>
Application Servers Home Page	<p>Use this Home Page to view a list of all the application server instances associated with a particular Oracle9iAS Infrastructure.</p>
Application Servers Home Page	<p>For more information, see "Using the Oracle9iAS Farm Home Page" on page 3-7.</p>
Application Servers Home Page	<p>The Farm Home Page is the first page you see if you have installed one or more application server instances that are associated with a common Oracle9iAS Infrastructure.</p>
Application Servers Home Page	<p>The Farm Home Page is also the first page you see when you view the Web site on a host where the Oracle9iAS Infrastructure has been installed.</p>
Application Servers Home Page	<p>Use this page to view a list of the application server instances installed on the current host. Click the name of an instance to drill down and display the Oracle9iAS Instance Home Page, or—if the instance you select is associated with an Oracle9iAS Infrastructure—click Farm to display the Oracle9iAS Farm Home Page.</p>
Application Servers Home Page	<p>This is the first Home Page you see if you have installed more than one application server instance on the host and one or more of the instances are not associated with an infrastructure.</p>

Using the Oracle9iAS Instance Home Page

From the Oracle9iAS Instance Home Page ([Figure 3–1](#)), you can start and stop the application server instance, monitor the overall performance of the server, and review the components of the server. You can also drill down and examine the performance of a particular component and configure the component.

Figure 3-1 Oracle9iAS Instance Home Page



If you scroll down the page, the Home Page provides a table that lists the components of the application server. From this table, you can also get a snapshot of how each individual component is performing.

From the System Components table, you can display a Home Page for each component of the application server.

Depending upon the options you have installed, the Administration section of the Oracle9iAS Instance Home Page provides additional features that allow you to perform the following tasks:

- Associate the current instance with an existing Oracle9iAS Infrastructure.
- Configure additional Oracle9iAS components that have been installed, but not configured
- Change the password or default schema for a component

For more information, click **Help** at the top of the Oracle9iAS Instance Home Page.

Using the Oracle9iAS Farm Home Page

If your application server instance is associated with an Oracle9iAS Infrastructure, your start page for the Oracle Enterprise Manager Web site is the Oracle9iAS Farm Home Page ([Figure 3-2](#)).

For more information about installing the Oracle9iAS Infrastructure, see the *Oracle9i Application Server Installation Guide*.

The Farm Home Page displays a list of the standalone application server instances and Oracle9iAS clusters associated with your Oracle9iAS Infrastructure. Standalone instances are application server instances that are not part of an Oracle9iAS cluster.

- You can associate a particular application server instance with the Infrastructure using the **Use Infrastructure** link on the Oracle9iAS Instance Home Page.

See Also: ["Associating an Instance with an Infrastructure \(Joining a Farm\)"](#) on page 10-4

Using the Farm Home Page, you can perform the following tasks:

- Manage multiple application server instances on multiple hosts
- Drill down to the Oracle9iAS Instance Home Page for each instance
- Create and manage Oracle9i Application Server clusters

See Also: [Chapter 14, "Application Server Clustering"](#) for more information about Oracle9iAS clusters

Figure 3–2 The Oracle9iAS Farm Home Page

The screenshot shows the Oracle Enterprise Manager Farm Home Page. At the top, there is a navigation bar with the Oracle logo, the text "Enterprise Manager", and links for "Targets", "Preferences", and "Help". Below the navigation bar, a blue header bar contains the text "Application Servers" and "Farm". Under "Farm", a link "Standalone Instances" is shown. The main content area is titled "Farm:myserver1.mycompany.com:1521:iasdb". A section titled "Clusters" displays a message: "All clusters belonging to farm "myserver1.mycompany.com:1521:iasdb" are listed below." It includes a "Create Cluster" button and navigation links for "Previous" and "Next". A table titled "Select Name" shows one row: "There are no clusters in the farm." A section titled "Standalone Instances" displays a message: "All instances that belong to farm "myserver1.mycompany.com:1521:iasdb" but are not part of any cluster are listed below." It includes a "Join Cluster" button and navigation links for "Previous" and "Next". A table titled "Select Name" lists two entries: "infra.myserver1.mycompany.com" and "ias1.myserver2.mycompany.com". The "Status Host" column shows green checkmarks for both, and the "Oracle Home" column shows "/private1/iasinst/install1" for the first entry and "/private1/iasinst/install3" for the second.

Using an Oracle9iAS Component Home Page

Oracle9iAS component Home Pages vary from one component to another. That is because each component has different monitoring and configuration requirements. However, most of the component Home Pages has the following common elements:

- A general information section that includes an icon that indicates the current state of the component and provides buttons for starting and stopping the component (if applicable).
- Status information, including CPU and memory usage information, so you can get a snapshot of how the component is performing.
- Component-specific information, such as a list of virtual hosts on the HTTP Server Home Page or a list of deployed applications on the OC4J Home Page.
- Links to administrative functions where appropriate, so you can modify the configuration of selected components. In many cases, this means you can use a graphical user interface to modify complex configuration files.

For more information about monitoring Oracle9iAS components, see [Chapter 9, "Monitoring HTTP Server, OC4J, and Web Cache"](#). For information about configuring Oracle9iAS components, see [Chapter 6, "Configuring HTTP Server, OC4J, and Web Cache"](#).

Using the Enterprise Manager Web Site Online Help

At any time while using the Enterprise Manager Web site, you can click **Help** at the top of the page to get more information. In most cases, the Help window displays a help topic about the current page. Click **Help Contents** in the Help window to browse the list of help of topics or to search for a particular word or phrase.

Starting and Stopping the Oracle Enterprise Manager Web Site

To use the Oracle Enterprise Manager Home Pages, you must start the Oracle Enterprise Manager Web site. The Web site is automatically started after you install the application server. You must start it manually after each system reboot, or create a script to automatically start it during system boot.

The following sections describe how to:

- Confirm that the required DISPLAY environment variable is set properly if you are using a UNIX system
- How to start, stop, and check the status of the Enterprise Manager Web site on UNIX and Windows systems

Setting the DISPLAY Environment Variable on UNIX Systems

Before you start the Oracle Enterprise Manager Web Site, you must be sure that the X Server software on your UNIX system is properly configured. Specifically, you must define the DISPLAY environment variable. This variable is required for the X Server software.

In most cases, the X Server software will be configured properly, but if you need to set the DISPLAY variable, use the following instructions:

1. Log in to the host that is running the Enterprise Manager Web Site.
2. Set the DISPLAY environment variable as follows:

If you are using ksh or sh, issue the following commands:

```
$ DISPLAY=hostname:0.0; export DISPLAY
```

If you are using csh or tcsh, issue the following commands:

```
% setenv DISPLAY hostname:0.0
```

In the examples, replace *hostname* with the name of a valid host running the X Server software.

3. If necessary, log in to the host you specified in Step 2 and make sure that X Server host you specified will allow remote access.

You allow remote access by entering the following command:

```
% xhost +
```

Starting and Stopping the Enterprise Manager Web Site on UNIX

If you need to start or stop the Oracle Enterprise Manager Web site on a UNIX system, use the commands shown in [Table 3–4](#).

Table 3–2 Starting and Stopping the Management Server

If you want to...	Enter the command...
Start the Enterprise Manager Web site	<code>emctl start</code>
Stop the Enterprise Manager Web site	<code>emctl stop</code> Enterprise Manager prompts you for the <code>ias_admin</code> management password. Note: If you are stopping the Enterprise Manager Web site because you are changing the <code>ias_admin</code> password, the password you enter here is the original password.
Verify the status of the Enterprise Manager Web site	<code>emctl status</code>

The `emctl` command is available in the `ORACLE_HOME/bin` directory after you install Oracle9iAS.

Note: If you have more than one Oracle home installed on your host computer, the Oracle home you installed first contains the active Oracle Enterprise Manager. The `emctl` command associated with the first Oracle home starts and stops the Web site on this host. To locate the active Oracle Enterprise Manager, view the contents of the file `/var/opt/oracle/emtab`.

You can verify the Enterprise Manager Web site is started by pointing your browser to the Web site URL. For example:

```
http://hostname:1810
```

Stopping the Enterprise Manager Web Site Process on UNIX

If you have trouble stopping the Enterprise Manager Web site, you can use the following procedure to locate and stop the Enterprise Manager Web site process on your UNIX system. For example, use this procedure if you have changed the `ias_admin` password and you do not have the original password:

1. Enter the following command at the UNIX command prompt to locate the process ID for the Enterprise Manager Web site:

```
$ /usr/ucb/ps auxww | grep EMDROOT
```

2. Note the process ID number that the command returns.
3. Use the following command to stop the process:

```
$ kill <process_number>
```

Starting and Stopping the Enterprise Manager Web Site on Windows

To start or stop the Enterprise Manager Web site on Windows, use the Services control panel. The name of the service is in the following format:

Oracle`ORACLE_HOME`EMwebsite

For example, if the name of the Oracle Home is `OraHome902`, the service name is:

OracleOraHome902EMWebsite

The Services control panel shows you the status of the service and indicates the method used to start the service. You can then select the service name and use the controls on the Services window to start and stop the service. When you stop the service, the Services control panel displays a dialog box asking for the `ias_admin` password.

You can also start or stop the service using the following command line:

```
net start WEB_SITE_SERVICE_NAME
```

For example:

```
net start OracleOraHome902EMWebsite
```

Changing the ias_admin Password

The `ias_admin` password is required to use the Enterprise Manager Web site. You can change the `ias_admin` user password in the following ways:

- [Using Oracle Enterprise Manager Web Site](#)
- [Using the emctl Command-Line Tool](#)

Warning: If you use an infrastructure, you must adhere to the Oracle Internet Directory password policy when setting the `ias_admin` password. This is because, even though the `ias_admin` password is not stored in Oracle Internet Directory, it may be used to set component passwords within Oracle Internet Directory. The default password policy is a minimum of five characters, with at least one numeric character. For more information see "[Password Policies](#)" on page 12-18.

Using Oracle Enterprise Manager Web Site

To change the `ias_admin` user password using Oracle Enterprise Manager Web site:

1. Navigate to the Instance Home Page. Select **Preferences** in the top right corner. This displays the Change Password Page.
2. Enter the new password and new password confirmation. Click **OK**. This resets the `ias_admin` password for all application server installations on the host.
3. Restart the Oracle Enterprise Manager Web site.

For more information, see "[Starting and Stopping the Oracle Enterprise Manager Web Site](#)" on page 9.

Using the emctl Command-Line Tool

To change the `ias_admin` user password using a command-line tool:

1. Enter the following command in the Oracle home of the primary installation (the first installation on the host):

```
(UNIX) $ORACLE_HOME/bin/emctl set password new_password  
(Windows) %ORACLE_HOME%\bin\emctl set password new_password
```

For example:

```
(UNIX) $ ORACLE_HOME/bin/emctl set password m5b8r5  
(Windows) %ORACLE_HOME%\bin\emctl set password m5b8r5
```

2. Restart the Enterprise Manager Web site.

For more information, see "[Starting and Stopping the Oracle Enterprise Manager Web Site](#)" on page 9.

Enabling Enterprise Manager Web Site Security

Oracle strongly recommends that you use the Secure Socket Layer (SSL) protocol and HTTPS for all connections to the Oracle Enterprise Manager Web site. There are two methods for setting up the Enterprise Manager Web site for secure connections:

- [Using SSL For Your Testing Environment](#)
- [Using SSL for Your Production Environment](#)

The following sections describe these methods in detail.

Note: The emctl command used in the following procedures is located in the following directory:

(UNIX) *ORACLE_HOME/bin*

(Windows) *ORACLE_HOME\bin*

Using SSL For Your Testing Environment

In your testing or development environment, you can enable SSL for the Enterprise Manager Web site without performing any significant configuration steps. However, this method of enabling security is based on a standard SSL Server Identity that is not trustworthy. Consider this method of setting up SSL if you are investigating SSL for the first time. This method of setting up security is not recommended for production systems.

To enable SSL for the Enterprise Manager Web site in your testing environment:

1. Stop the Enterprise Manager Web site.

For more information, see "[Starting and Stopping the Oracle Enterprise Manager Web Site](#)" on page 3-9

2. Enter the following emctl command:

```
emctl set ssl test
```

3. Restart the Enterprise Manager Web site.
4. Connect to the Web site using the following HTTPS address:

`https://server_name:1810`

To disable SSL in your testing environment:

1. Stop the Enterprise Manager Web site.

For more information, see "[Starting and Stopping the Oracle Enterprise Manager Web Site](#)" on page 3-9.

2. Enter the following `emctl` command:

`emctl set ssl off`

3. Restart the Enterprise Manager Web site.

4. Connect to the Enterprise Manager Web site using the following unsecured address:

`http://server_name:1810`

Using SSL for Your Production Environment

For your production systems, Oracle strongly recommends that you set up security for the Enterprise Manager Web site using a valid security certificate. For more information about security certificates, see the *Oracle9i Application Server Security Guide*.

Note: Perform each of the steps in this section in the order specified. Do not repeat the same step multiple times or skip any steps. You may repeat this procedure any number of times provided that you start at Step 1 each time.

To set up SSL for Enterprise Manager Web site in your production environment:

1. Generate a Request for a new SSL Server Site Certificate by entering the following command:

`emctl gencertrequest`

Enterprise Manager generates a Certificate Request, stores the request in a file called `server.csr`, and shows you the location of this file. The content of the file looks similar to the output shown in [Example 3-1](#).

2. Copy the Certificate Request text, paste it into an e-mail message, and send the message to a certificate authority.

For more information about certificate authorities, see the *Oracle9i Application Server Security Guide*.

The Certificate Authority returns to you two certificate files. One is the Certificate Authority Certificate and the other is the SSL Server Site Certificate. The content of each Certificate file looks similar to the one shown in [Example 3-2](#).

3. Install the Certificate Authority Certificate, as follows:
 - a. Save the Certificate Authority Certificate and note the location of the file.
 - b. Enter the following command:

```
emctl installcert -ca certificate_authority_certificate_path
```

For example:

```
emctl installcert -ca /home/myfiles/oacertificate.cer
```

If the certificate is installed successfully, the following message appears:

```
OC4J keystore was updated at  
$ORACLE_HOME/sysman/j2ee/server/keystore.secure
```

4. Install the SSL Server Site Certificate, as follows:

- a. Save the Server Site Certificate and note the location of the file.
 - b. Enter the following command:

```
emctl installcert -cert SSL_Server_Site_Certificate_path
```

For example:

```
emctl installcert -cert /home/myfiles/ssl_certificate.cer
```

If the SSL Site Certificate is installed successfully, the following message appears:

```
OC4J keystore was updated at  
$ORACLE_HOME/sysman/j2ee/server/keystore.secure
```

OC4J keystore is ready for SSL.

5. Password protect your new SSL Server Site Certificate.

Steps 1 through 4 establish the material needed for the Enterprise Manager Web site to act as a unique and trustworthy SSL Server Site. Oracle recommends that you protect this material with a password. The initial default password is "welcome". To modify the password, enter the following command:

```
emctl set ssl password old_password new_password
```

For example:

```
emctl set ssl password welcome manchester123
```

6. Enable SSL for the Enterprise Manager Web site as follows:

a. Stop the Enterprise Manager Web site.

For more information, see "[Starting and Stopping the Oracle Enterprise Manager Web Site](#)" on page 3-9.

b. Enter the following emctl command:

```
emctl set ssl test
```

c. Restart the Enterprise Manager Web site.

d. Connect to the Web site using the following HTTPS address:

```
https://server_name:1810
```

Example 3-1 Sample Certificate Request

```
-----BEGIN NEW CERTIFICATE REQUEST-----
MIIByzCCARACAAQAwZzELMAkGA1UEBhMCVVMxEzARBgNVBAgTCkNhG1mb3JuaWExDDAKBgNVBAcT
A0VNRDEMMa0GA1UEChMDRU1EMQwwCgYDVQQLEwNFTUQxGTAXBgNVBAMTEFRFU1QgQ0VSVE1GSUNB
VEUwgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAM1saVui7S/f+bFLBmP2nFhHcAf/RNMViWg
5-MRTBBC1anSPTtABle54VOrTEXPACbmedD1eMbr+585=wjC+MHaJoEtEXSUvRoYin4OP-XOX-H
I2GfwYnYf4G6ZmbvopKPYx9NxfKdqathbiR8ayG+TH92Yso0Jdsyry2nnDltAgMBAAGgADANBgkq
hkiG9w0BAQQFAAOBgQC1ZmkghQUjmFMaFCx2+wMZQNpQ0AQEtaz3MNQeChIIIVkZLeGgRoZ0g4HF
CL1SOp2J12+dKuW4N+xa9y9Vo0vVtAAy717a4r83CyotaMOLXW70YuxRxzs6a30xwlH1AUFr3KUY
xyr/Ij0xYRyMOnk4INLK6EC1ght+BnYHo77imw==
-----END NEW CERTIFICATE REQUEST-----
```

Example 3–2 Sample SSL Certificate

```
-----BEGIN CERTIFICATE-----
MIIB4jCCAYygAwIBAgIQUcmat1ccRQmg8ZuXwne+qTANBgkqhkiG9w0BAQQFADA5
MR4wHAYDVQQKFBVPcmFjbGUgU2VjcmV0IFNlcnPzY2UxZAVBqNVBAMUDkR1dmVs
b3BtZW50IENBMB4XDIAyMDIwNTIxMzM1MVoXDTAyMDgwNDIxMzM1MVoWZzELMAkG
A1UEBhMCVVMxEzARBgNVBAgTCkNhbgLmb3JuaWExDDAKBgNVBAcTA0VNREDEMMACG
A1UEChMDRU1EMQwwCgYDVQQLEwNFTUQxGTAXBqNVBAMTEFRFU1QgQ0VSVE1GSUNB
VEUwgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAM1saVui7S/f+bFLMBmP2nFh
HcAf/RNMViWgpKgCsN8JyFJXwLOIqd8i+Ggf0VT9wfGkI4+8oSIVdhiLJdZ431VB
agbDGpf03nmj4NNCnxNkrG+yI2GfwYnYf4G6ZmbvopKPYx9NxfKdqathbiR8ayG+
TH92YsoJDsyry2nnDltAgMBAAEwDQYJKoZIhvcNAQEEBQADQQCcGj+eRsUrg6Pt
typn6q25ZlfpZw3VMLqf8wQ+uEjGw6Y+Bd5f7tAoPXsjxdzVdB2GG+eHZI0eipA9
VIRtwZM1
-----END CERTIFICATE-----
```

About the Underlying Technologies

The Enterprise Manager Web site relies on various technologies to discover, monitor, and administer the Oracle9iAS environment.

These technologies include:

- Oracle Dynamic Monitoring Service (DMS)

The Enterprise Manager Web site uses DMS to gather performance data about your Oracle9iAS components. To learn more about DMS, see the *Oracle9i Application Server Performance Guide*.

- Oracle HTTP Server and Oracle Containers for J2EE (OC4J)

While you can use Oracle HTTP Server and OC4J to deploy your own applications, the Enterprise Manager Web site also uses HTTP Server and OC4J to deploy its management components. When you start the Web site for the first time, you will notice that several default OC4J instances are reserved for Oracle management applications. For more information, see the topic "About the Default OC4J Instances" in the Enterprise Manager Web site online help.

- Oracle Process Management Notification (OPMN)

OPMN manages Oracle HTTP Server and OC4J processes within an application server instance. It channels all events from different component instances to all components interested in receiving them.

See Also: [Chapter 13, "High Availability"](#) for more information about OPMN

- **Distributed Configuration Management (DCM)**

DCM manages configurations among application server instances that are associated with a common Infrastructure (members of an Oracle9iAS farm). It enables Oracle9iAS cluster-wide deployment so you can deploy an application to an entire cluster, or make a single host or instance configuration change applicable across all instances in a cluster. The Enterprise Manager Web site uses DCM to make configuration changes and to propagate configuration changes and deployed applications across the cluster.

See Also: [Chapter 14, "Application Server Clustering"](#) and [Appendix F, "DCM Command-Line Utility \(dcmctl\)"](#) for more information on DCM

About the Oracle Enterprise Manager Console

The Oracle Enterprise Manager Web site provides all the tools you need to manage your application server instances, farms, clusters, and system components.

However, if you have an environment that includes other Oracle products and applications in addition to Oracle9iAS, consider using the Oracle Enterprise Manager Console.

The Oracle Enterprise Manager Console, when used with the Oracle Enterprise Manager Web site, provides a wider view of your Oracle environment beyond Oracle9iAS. Use the Console to automatically discover and manage databases, application servers, and Oracle applications across your entire network.

The Console offers advanced management features, such as an Event system to notify administrators of changes in your environment and a Job system to automate standard and repetitive tasks, such as executing a SQL script or executing an operating system command.

The following sections describe:

- [The Enterprise Manager Three-Tier Framework](#)
- [Installing the Framework](#)
- [Configuring the Management Server](#)
- [Starting and Stopping Oracle Management Server on UNIX](#)
- [Using the Console](#)
- [Starting and Stopping the Intelligent Agent on UNIX](#)
- [Obtaining More Information](#)

The Enterprise Manager Three-Tier Framework

The Oracle Enterprise Manager Console is part of the Oracle Enterprise Manager three-tier framework. Unlike a two-tier client-server structure, where a client tier is responsible for the presentation of information and accesses data from a server, Oracle Enterprise Manager utilizes a three-tier architecture:

- Console clients and integrated tools provide a graphical interface for administrators--first tier
- Management Servers and a database repository provide a scalable middle tier for processing system management tasks--second tier
- Intelligent Agents installed on each host computer monitor its services and execute tasks from the Management Server--third tier

For a more complete description of the Oracle Enterprise Manager framework, see the *Oracle Enterprise Manager Configuration Guide*.

Installing the Framework

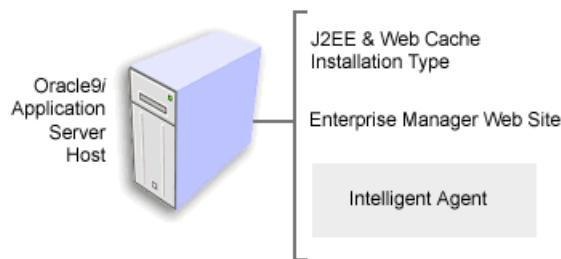
The following sections describe how to install the Enterprise Manager three-tier framework so you can use the Console to manage your Oracle environment.

Installing the Intelligent Agent

The Oracle Intelligent Agent is installed whenever you install Oracle9iAS on a host computer. For example, if you select the J2EE and Web Cache installation type, the Oracle Universal Installer installs Oracle Enterprise Manager Web site and the Oracle Intelligent Agent, along with the J2EE and Web Cache software ([Figure 3-3](#)). This means the Intelligent Agent software is always available if you decide to use the Console and the Management Server to manage your Oracle9iAS environment.

For more information, see "[Starting and Stopping the Intelligent Agent on UNIX](#)" on page 3-21.

Figure 3–3 Intelligent Agent Installed on an Application Server Host

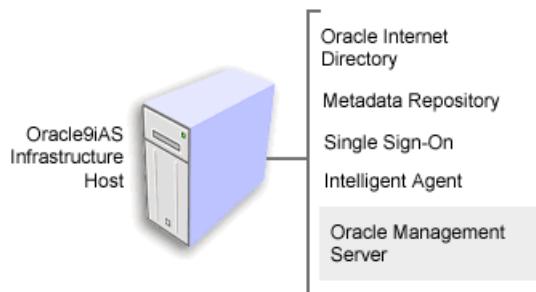


Installing the Console and Management Server

The Console and Management Server are installed as part of the Oracle9iAS Infrastructure. In most cases, you install the Infrastructure on a dedicated host that can be used to centrally manage multiple application server instances. The Infrastructure includes Oracle Internet Directory, Single Sign-On, the metadata repository, the Intelligent Agent, and Oracle Management Server (Figure 3–4).

Note: The Console and the Management Server are installed as part of the Infrastructure installation, but the Management Server will not be configured for use unless you select **Oracle Management Server** on the Select Configuration Options page. For more information, see the *Oracle9i Application Server Installation Guide*.

Figure 3–4 Oracle Management Server on an Infrastructure Host



Starting and Stopping the Intelligent Agent on UNIX

On UNIX systems, use the `agentctl` utility to start and stop the Agent. The `agentctl` command is available in the `ORACLE_HOME/bin` directory after you install the Oracle9iAS Infrastructure.

The relevant `agentctl` commands are listed in [Table 3-3](#).

Table 3-3 Starting and Stopping the Intelligent Agent

If you want to...	Enter the command...
Start the Agent	<code>agentctl start agent</code>
Stop the Agent	<code>agentctl stop agent</code>
Verify the status of the Agent	<code>agentctl status agent</code>

Starting and Stopping the Intelligent Agent on Windows

On Windows systems, use the Services control panel to start and stop the Agent. The name of the service is in the following format:

`OracleORACLE_HOMEAgent`

For example:

`OracleOraHome902Agent`

The Services control panel shows you the status of the service and indicates the method used to start the service. You can then select the service name and use the controls on the Services window to start and stop the service.

You can also start or stop the service using the following command line:

`net start AGENT_SERVICE_NAME`

For example:

`net start OracleOraHome902Agent`

For more information about configuring and controlling the Intelligent Agent, see the *Oracle Intelligent Agent User's Guide*.

Configuring the Management Server

The Oracle Enterprise Manager Configuration Assistant is launched as a configuration tool at the end of the Oracle9iAS installation. The configuration

assistant runs silently in the background and automatically configures Oracle Management Server to use the Metadata Repository of the Infrastructure you are installing.

Note: The Console and the Management Server are installed as part of the Infrastructure installation, but the Management Server will not be configured for use unless you select **Oracle Management Server** on the Select Configuration Options page. For more information, see the *Oracle9i Application Server Installation Guide*.

If you do not configure the Management Server as part of the Infrastructure installation, you can run the Oracle Enterprise Manager Configuration Assistant later. When you run the Assistant later, you can configure Oracle Management Server so it uses the metadata repository, or you can configure the server to use another database on your network.

The following sections describe:

- The capabilities of the Enterprise Manager Configuration Assistant
- How to start the Configuration Assistant
- How to get help while using the Configuration Assistant
- How to manually configure the Management Server so it uses the Oracle9iAS metadata repository

About the Oracle Enterprise Manager Configuration Assistant

The Oracle Enterprise Manager Configuration Assistant guides you through the steps to configure Oracle Management Server. For example, you use the Configuration Assistant to:

- Create a database instance for the Enterprise Manager repository or select an existing database instance for the repository.
- Create or select an OEM_REPOSITORY tablespace.
- Create a new database user who will own the repository, specifying default and temporary tablespaces for the user and assigning proper privileges and roles to the user.

For a complete description of the tasks performed by Oracle Enterprise Manager Configuration Assistant, see the *Oracle Enterprise Manager Configuration Guide*.

Starting the Configuration Assistant

To start the Configuration Assistant at any time, enter the following command:

```
emca
```

The `emca` command is available in the following directory after you install Oracle9iAS Infrastructure:

```
(UNIX) ORACLE_HOME/bin  
(Windows) ORACLE_HOME\bin
```

Getting Help While Using the Configuration Assistant

The *Oracle Enterprise Manager Configuration Guide* contains detailed instructions about running the Oracle Enterprise Manager Configuration Assistant and configuring Oracle Management Server.

In addition, click **Help** at any time while running the Configuration Assistant to display the Oracle Enterprise Manager online help. Use the icons in the Help window toolbar to display the Enterprise Manager online help table of contents or to search for a word or phrase in the online help.

Manually Configuring the Oracle Management Server

When you install the Oracle9iAS Infrastructure, the installation procedure automatically creates a default schema, user, and password for the Enterprise Manager repository. When you run the Enterprise Manager Configuration Assistant, you can choose to use this existing repository, or you can create a new repository.

The following instructions describe how to manually configure the management server so it uses the default schema, user, and password:

1. Start the Enterprise Manager Configuration Assistant and select the following options:
 - Configure Local Management Server
 - Use Existing RepositoryWhen you select these options, the Configuration Assistant creates the `omsconfig.properties` file, which is a file required by the Oracle Management Server.
2. Next, provide the following default information for the repository:
 - Repository Username: OEM_REPOSITORY

- Repository Password: `oem_repository`
- Connection Information: `<host:port:sid>`

`OEM_REPOSITORY` is the default Enterprise Manager schema. When you installed Oracle9iAS infrastructure, the `OEM_REPOSITORY` database user was created and the necessary Enterprise Manager schema was loaded. Oracle recommends that you do not change this user. You can however, change the default password. The repository username and password are stored in the `omsconfig.properties` file. The configuration assistant will encrypt the repository password in this file.

The connection information includes the name of the host where the metadata repository resides, the port for the database (usually 1521), and the system identifier (SID) for the database. For example:

```
mysystem:1521:iasdb
```

Starting and Stopping Oracle Management Server on UNIX

To start or stop Oracle Management Server on a UNIX system, use the commands shown in [Table 3–4](#).

Table 3–4 Starting and Stopping the Management Server

If you want to...	Enter the command...
Start Oracle Management Server	<code>oemctl start oms</code>
Stop Oracle Management Server	<code>oemctl stop oms <username/password></code> If you do not provide the user name and password on the command line, the management server prompts you for the user name and password. The default user name is <code>sysman</code> . The default password is <code>oem_temp</code> .
Verify if Oracle Management Server is running or not	<code>oemctl ping oms</code>
Verify the status of Oracle Management Server	<code>oemctl status oms</code>

The `oemctl` command is available in the `ORACLE_HOME/bin` directory after you install the Oracle9iAS Infrastructure.

For complete instructions on obtaining the status of the Management Server, see the *Oracle Enterprise Manager Configuration Guide*.

Starting and Stopping the Oracle Management Server on Windows

On Windows systems, use the Services control panel to start and stop the management server. The name of the service is in the following format:

```
OracleORACLE_HOMEManagementServer
```

For example:

```
OracleOraHome902ManagementServer
```

The Services control panel shows you the status of the service and indicates the method used to start the service. You can then select the service name and use the controls on the Services window to start and stop the service. When you stop the management server, the Services control panel displays a dialog box that prompts you for the user name and password. The default user name is `sysman`. The default password is `oem_temp`.

You can also start or stop the service using the following command line:

```
net start AGENT_SERVICE_NAME
```

For example:

```
net start OracleOraHome902ManagementServer
```

For more information about configuring and controlling the management server, see the *Oracle Enterprise Manager Configuration Guide*.

Using the Console

The following sections describe how to get started using Oracle Enterprise Manager Console.

Starting the Console

After you have configured and started Oracle Management Server, you can start the Oracle Enterprise Manager Console. To start the console:

1. Display the Enterprise Manager Login dialog box as follows:

- (UNIX) Enter the following command:

```
oemapp console
```

The `oemapp` command is available in the following directory after you install the Oracle9iAS Infrastructure:

ORACLE_HOME/bin

- (Windows) From the **Start** menu, select **Oracle - ORACLE_HOME**, then click **Enterprise Manager Console**.
- 2. On the Login dialog box, select **Login to the Oracle Management Server**.
- 3. Enter the Enterprise Manager administrator name and password and the Management Server host name.

The default Enterprise Manager administrator name is `sysman` and its password is `oem_temp`.

The Management Server host that you specify in the Login dialog box must already be configured with the repository you want to access. To select an existing Management Server, click the icon next to the **Management Server** field.

- 4. Click **OK**.

If this is the first time you are starting the Console, Enterprise Manager displays a dialog asking you for a new password to replace the default `oem_temp` password. Enter a new password and click **OK**.

Enterprise Manager displays the Console main window.

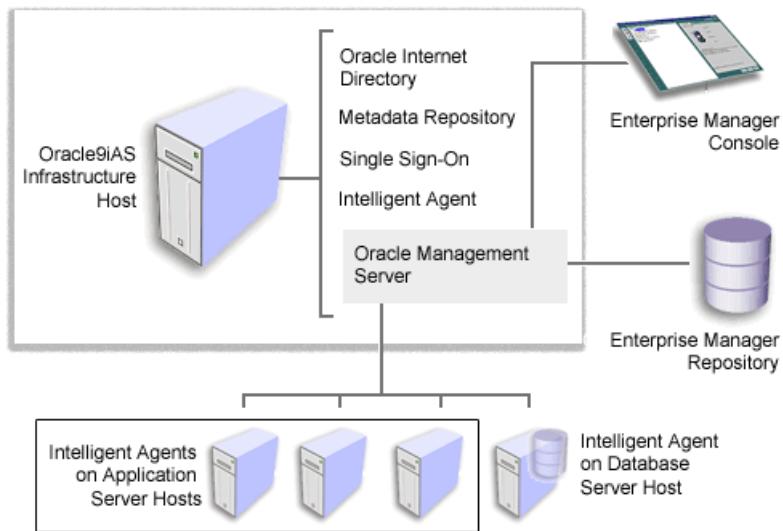
Discovering Targets

Oracle Enterprise Manager provides a Discovery Wizard for identifying hosts and populating the Console Navigator tree with these discovered hosts and the targets on those hosts. The discovered targets, such as databases and listeners, can then be administered with Enterprise Manager.

During startup of the Console, any manageable targets on the machine where the Management Server is running are automatically discovered if the Intelligent Agent is installed and running on that Management Server machine. The Console Navigator then displays all those discovered targets. For more information, see "[Starting and Stopping the Intelligent Agent on UNIX](#)" on page 3-21.

To start the Discovery Wizard, choose **Discover Nodes** from the **Navigator** menu.

Figure 3–5 Overview of a Working Oracle Management Server in Oracle9iAS



For more information about using the Console to manage Oracle9iAS, see ["Using the Console with the Oracle Enterprise Manager Web Site"](#) on page 3-28.

Using Jobs and Events

Besides offering you a wider view of your Oracle9iAS environment, the Console also provides two important features for administrators: jobs and events.

The Jobs system allows you to automate standard and repetitive tasks, such as executing a SQL script or executing an operating system command. With the Job system, you can create and manage jobs, share jobs with other administrators, schedule execution of jobs, and view information about the jobs. Jobs can be scheduled on a single node or multiple nodes in the network, provided that the node has an Intelligent Agent running on it.

The Event system allows you to monitor your network for specific conditions, such as loss of service or lack of storage, that may occur in your managed environment. You select tests to run on managed targets (databases, nodes, listeners, or other services), then set the threshold parameters for which you want to be notified. You can share events with other administrators, in addition to being able to notify specific administrators when an event condition occurs. For some event tests, you can also choose to execute a fixit job that automatically corrects the problem.

The Console includes several events that are specific to Oracle9iAS. For more information about the Oracle9iAS events, see the *Oracle Enterprise Manager Addendum to Event Test Reference Manual*.

To learn more about creating jobs and events using the Console, see the *Oracle Enterprise Manager Administrator's Guide*.

Obtaining More Information

For information about starting, configuring, and using the Oracle Enterprise Manager Console, see the following documentation on your Oracle9i Application Server Documentation Library CD-ROM:

- *Oracle Enterprise Manager Concepts Guide*
- *Oracle Enterprise Manager Administrator's Guide*
- *Oracle Enterprise Manager Configuration Guide*
- *Oracle Intelligent Agent User's Guide*

The Console also provides extensive online help. To display the Console online help, click the **Help** button on a particular dialog box or property page, or select a command from the Console **Help** menu.

Using the Console with the Oracle Enterprise Manager Web Site

The Console and the Oracle Enterprise Manager Web site are designed to work together. For example, suppose you use the Console to discover a host computer where Oracle9iAS has been installed. As long as the Intelligent Agent on that host is up and running, the application server instances on that host appear in the Console navigator pane.

To display the Enterprise Manager Home Page for an application server instance:

1. Select an application server instance in the Console navigator.

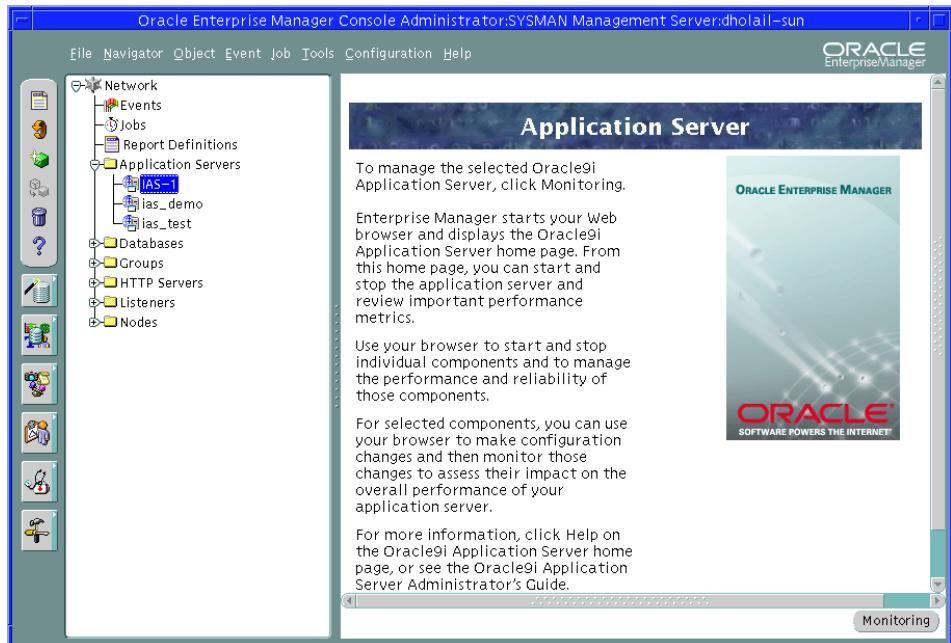
The Console displays an information panel describing the Oracle9iAS management tools ([Figure 3–6](#)).

2. Click Monitoring at the bottom of the information panel.

Oracle Enterprise Manager opens your default browser and displays the appropriate Oracle Enterprise Manager Home Page for the selected application server instance.

Note: To display the Enterprise Manager Home Page, you must have a default browser identified in the PATH environment variable.

Figure 3–6 Oracle Enterprise Manager Console



4

Postinstallation Tasks

This chapter describes the tasks you should perform after every application server installation.

It contains the following topics:

- [Task 1: View the Oracle9iAS Welcome Page](#)
- [Task 2: Check for Port Conflicts](#)
- [Task 3: Complete Component Configuration](#)
- [Task 4: Run Demos \(optional\)](#)

Task 1: View the Oracle9iAS Welcome Page

The Oracle9iAS Welcome Page is the entry point for managing the application server. It provides links to documentation, demos, management tools, and new features. Figure 4-1 displays the Oracle9iAS Welcome Page.

Figure 4-1 Oracle9iAS Welcome Page

Welcome

to Oracle9i Application Server

Documentation

The Oracle9iAS documentation set consists of the documentation library and the platform-specific documentation.

- The Oracle9iAS documentation library contains administration, configuration, and development documentation. The documentation library is on its own CD-ROM in the Oracle9iAS CD pack.
- The Oracle9iAS platform-specific documentation includes installation and migration documentation and release notes. The platform-specific documentation is on the Oracle9iAS Disk 1 CD-ROM in the Oracle9iAS CD pack.

The Oracle9iAS documentation is also available from Oracle Technology Network at <http://otn.oracle.com/products/ias>.

Quick Tour

The Quick Tour introduces the features and functionality of Oracle9iAS. You can find the Quick Tour in the Oracle9iAS documentation library, which is on a separate CD-ROM in the Oracle9iAS CD pack. You can also access the Quick Tour from Oracle Technology Network at <http://otn.oracle.com/products/ias>.

Release Notes
Read the [Release Notes](#) for important information about Oracle9iAS Release 2.

Oracle Enterprise Manager
Log on to [Oracle Enterprise Manager](#) to manage and monitor Oracle9iAS.

New Features

- [J2EE and Internet Applications](#)
- [Portal](#)
- [Wireless](#)
- [Caching](#)
- [Business Intelligence](#)
- [E-Business Integration](#)
- [Management and Security](#)

For details, visit Oracle Technology Network at <http://otn.oracle.com/products/ias>.

You can access the Welcome Page by pointing your browser to the HTTP Server URL for your installation. For example, the default HTTP Server URL is:

`http://hostname:7777`

Task 2: Check for Port Conflicts

The application server automatically assigns port numbers for many services during installation. It chooses port numbers from a set of default port numbers and port ranges.

See Also: [Appendix D, "Default Port Numbers and Port Ranges"](#)

A port should only be assigned to a single service. If two services attempt to use the same port, there will be a port conflict, and one or both services will not function properly.

When assigning ports, the application server attempts to avoid port conflicts in the following ways:

- It internally tracks the port numbers used by all application server instances (running or not running) on a host and will not use the same port number twice.
- It performs a system check to determine if any *non*-Oracle9iAS processes are running on the port.

Despite this strategy, a port conflict may still occur between the application server and a *non*-Oracle9iAS process that was not running during the application server installation.

Oracle recommends that you review the application server port numbers assigned during installation to determine if there are any port conflicts with the rest of your system. You can view application server port numbers in the following ways:

- Point your browser to the Oracle9iAS Welcome Page and select the **Ports** tab. You can access the Welcome Page using the HTTP Server URL, for example:

`http://host:7777`

- View the following file:

(UNIX) `ORACLE_HOME/install/portlist.ini`.
(Windows) `ORACLE_HOME\install\portlist.ini`

If you determine there is a port conflict, you must change the port number used by the application server or the *non*-Oracle9iAS process.

See Also: Component administration guides for information on changing port numbers for specific components

Task 3: Complete Component Configuration

If you configured any of these components during installation, you may need to perform some additional steps to complete the configuration:

- [Oracle9iAS Clickstream Intelligence](#)
- [Oracle9iAS Discoverer](#)
- [Oracle9iAS Personalization](#)
- [Oracle9iAS Portal](#)
- [Oracle9iAS Unified Messaging](#)

If you would like to use mod_plsql, it was installed with Oracle HTTP Server but requires some additional steps for configuration.

This section describes the steps to complete the configuration of each component.

Oracle9iAS Clickstream Intelligence

After completing a Business Intelligence and Forms or Unified Messaging installation, you can configure Oracle9iAS Clickstream Intelligence to use the metadata repository or a customer database (also referred to as a *dedicated Clickstream database*). If you plan to acquire and store a large volume of Web log data, it is recommended that you use a customer database.

See Also: To configure Clickstream to use a customer database, follow the steps in Appendix B, "Installing a Dedicated Clickstream Database" in *Oracle9iAS Clickstream Intelligence Administrator's Guide*.

See Also: To configure Clickstream to use a metadata repository, follow the steps in Appendix E, "Metadata Repository Postinstallation and Configuration" in *Oracle9iAS Clickstream Intelligence Administrator's Guide*.

Oracle9iAS Discoverer

If you configured Oracle9iAS Discoverer during installation, you must install Discoverer workbooks and End User Layer (EUL) into each customer database that contains the data to be analyzed.

See Also: *Oracle9i Discoverer Administrator Administration Guide* for information about preparing a customer database for Discoverer to use

Oracle9iAS Personalization

If you configured Oracle9iAS Personalization during installation, you must load schemas into a customer database so they can be co-resident with the Personalization data. You can do this using the Oracle Personalization Schema Creation Wizard.

See Also: *Oracle9iAS Personalization Administrator's Guide* for information on running the Oracle Personalization Schema Creation Wizard

Oracle9iAS Portal

If you configured Oracle9iAS Portal during the application server installation, it was configured to use the metadata repository, and no additional steps are required.

If you did not configure Oracle9iAS Portal during installation, you can configure it any time after installation by using the Oracle Portal Configuration Assistant. When you configure Portal after installation, you can use the metadata repository or a customer database.

See Also: *Oracle9iAS Portal Configuration Guide* for information on configuring Oracle9iAS Portal using Oracle Portal Configuration Assistant

Oracle9iAS Unified Messaging

If you configured Oracle9iAS Unified Messaging during installation, you must install the mail store and middle-tier servers by running the `umconfig.sh` script that is located in `ORACLE_HOME/oes/bin`.

See Also: *Oracle9iAS Unified Messaging Administrator's Guide* for information on configuring the Unified Messaging system

mod_plsql

The mod_plsql module is installed as part of Oracle HTTP Server. In order to use mod_plsql, you must perform the following steps:

1. Install the PL/SQL Web Toolkit into a database.

The method for installing the PL/SQL Web Toolkit into a database depends on the type of database, as described in [Table 4-1](#).

Table 4-1 Methods for Installing PL/SQL Web Toolkit into a Database

Database Type	Method for Installing PL/SQL Web Toolkit
Metadata Repository (Oracle9iAS Portal was <i>not</i> configured during installation)	You must manually install the PL/SQL Web Toolkit using the <i>owaload.sql</i> script. See Also: <i>mod_plsql User's Guide</i>
Metadata Repository (Oracle9iAS Portal was configured during installation)	You do not need to install the PL/SQL Web Toolkit because it was already installed for Oracle9iAS Portal.
Customer Database with no prior Oracle9iAS Portal or PL/SQL packages	You must manually install the PL/SQL Web Toolkit using the <i>owaload.sql</i> script. See Also: <i>mod_plsql User's Guide</i>
Customer Database configured for Oracle9iAS Portal	You do not need to install the PL/SQL Web Toolkit because it was already installed for Oracle9iAS Portal.
Customer Database with pre-9.0.2 Oracle9iAS version of PL/SQL Web Toolkit, or WebDB	You must upgrade the PL/SQL Toolkit and perform some migration tasks. See Also: <i>mod_plsql User's Guide</i> and <i>Migrating from Oracle9iAS Release 1 (1.0.2.2.x) to Release 2 (9.0.2)</i>
Customer Database with Oracle Application Server (OAS) PL/SQL Web Toolkit	You must upgrade the PL/SQL Toolkit and perform some migration tasks. See Also: <i>mod_plsql User's Guide</i> and <i>Migrating from Oracle Application Server</i>

2. Create a Database Access Descriptor (DAD).

You can create a Database Access Descriptor (DAD) using Oracle Enterprise Manager.

See Also: ["Creating a Database Access Descriptor \(DAD\) for mod_plsql"](#) on page 6-11.

Task 4: Run Demos (optional)

Oracle9iAS provides demonstration software for many components, as described in [Table 4–2, "Oracle9iAS Demos"](#).

In order to run a demo for a particular component, you must have the following:

- An application server installation that contains the component
- The component must be configured
- The required component instances must be running

For example, the Portal demonstration will work only with the Portal and Wireless install type, or higher, Portal must be configured and the Portal, OC4J_Portal, and OC4J_Demos instances must be running.

See Also: To view more information and run demos, point your browser to the Oracle9iAS Welcome Page and select the **Demonstrations** tab.

Table 4–2 Oracle9iAS Demos

Component	Required Running Component Instances	Demo	Description
Oracle HTTP Server	HTTP Server	CGI	Displays environment setting using CGI.
	HTTP Server	Perl	Displays environment settings using a Perl script and mod_perl.
	HTTP Server	FastCGI	Displays environment settings using FastCGI.

Table 4–2 Oracle9iAS Demos (Cont.)

Component	Required Running Component Instances	Demo	Description
Oracle9iAS Containers for J2EE (OC4J)	OC4J_home OC4J_Demos	JSP	Demonstrates the Oracle JSP Tag Libraries, JDBC access, and Java Edge Side Includes (JESI).
	OC4J_home OC4J_Demos	JSP	Standard JSP examples from Sun.
	OC4J_home OC4J_Demos	Servlets	Standard servlet examples from Sun.
	OC4J_home OC4J_Demos	Java Object Cache	Demonstrates performance gains by caching Java objects.
	OC4J_home OC4J_Demos	BC4J	Demonstrates an online order system built with BC4J, with pages to browse, search, and order toys.
	OC4J_home OC4J_Demos	XML-based JAAS	Demonstrates the XML-based JAAS Provider to authenticate a user and enforce associated roles and permissions.
	OC4J_home OC4J_Demos	JDBC	Run these JDBC examples from the command line.
	OC4J_home OC4J_Demos	JMS	Run these Oracle Advanced Queuing JMS examples from the command line.
Oracle9iAS Web Services	OC4J_home OC4J_Demos	Web Services	Demonstrates Web Services.
Oracle9iAS Developer Kits	OC4J_home OC4J_Demos	XSQL Pages	Demonstrates examples of XML database publishing using XSLT for styling the output for delivery to the browser.
Oracle9iAS Web Cache	Web Cache	Web Cache Invalidation APIs	Provides sample code for Web Cache Invalidation APIs.
	Web Cache	Web Cache	Links to the Web Cache page on OTN for additional sample code.

Table 4–2 Oracle9iAS Demos (Cont.)

Component	Required Running Component Instances	Demo	Description
Oracle9iAS Portal	Portal	Portal	Links to a Portal page which includes a sample document repository and links to Oracle Portal Center and Oracle9iAS Portal Developer Kit.
	OC4J_Portal		
	OC4J_Demos		
Oracle9iAS Wireless	Wireless	Wireless	Demonstrates sending messages and accessing email from any mobile device.
	OC4J_Wireless		
	OC4J_Demos		
Oracle9iAS Discoverer	Discoverer	Discoverer	Demonstrates Discoverer Viewer and Discoverer Plus.
	OC4J_BI_Forms		
	OC4J_Demos		
Oracle9iAS Reports Services	Reports	Reports Services	Demonstrates dynamic report output to both paper and the Web.
	OC4J_BI_Forms		
	OC4J_Demos		
Oracle9iAS Forms Services	Forms	Forms Services	Demonstrates a test form to verify your install. You will be prompted to install JInitiator on the client as part of this demo.
	OC4J_BI_Forms		
	OC4J_Demos		
	Forms	Forms Services	Links to the Forms page on OTN for additional demos and sample code.
	OC4J_BI_Forms		
	OC4J_Demos		

Task 4: Run Demos (optional)

5

Starting and Stopping the Application Server

This chapter describes how to start and stop application server instances, infrastructures, and customer databases. It also provides information about starting and stopping individual components.

It contains the following topics:

- [Introduction](#)
- [Starting and Stopping Application Server Instances](#)
- [Starting and Stopping Components](#)
- [Starting an Infrastructure](#)
- [Stopping an Infrastructure](#)

Introduction

An application server enterprise contains application server instances, infrastructures, and customer databases that can be started and stopped. Most commonly you will start and stop these pieces of your enterprise separately, for example, when rebooting a system. You may also want to start and stop your entire enterprise, for example, to reconfigure it or perform routine maintenance.

This section provides high-level information about starting and stopping an entire enterprise and its individual pieces. The rest of this chapter provides detailed procedures for starting and stopping application server instances and infrastructures, as well as starting and stopping individual components within instances.

Starting an Application Server Enterprise

The order in which to start the pieces of an application server enterprise is as follows:

1. Start the infrastructure.

If your enterprise contains more than one infrastructure, start the primary infrastructure first.

2. Start customer databases.

If your enterprise contains customer databases, you can start them using several methods, including SQL*Plus and Oracle Enterprise Manager Console.

See Also: Oracle database administration guides and *Oracle Enterprise Manager Administrator's Guide* for more information about starting databases

3. Start application server instances.

You can start application server instances in any order. If instances are part of a cluster, start them as part of starting the cluster.

See Also: [Chapter 14, "Application Server Clustering"](#) for information about clusters

Stopping an Application Server Enterprise

The order in which to stop the pieces of an application server enterprise is as follows:

1. Stop application server instances.

You can stop application server instances in any order. If instances are part of a cluster, stop them as part of stopping the cluster.

See Also: [Chapter 14, "Application Server Clustering"](#) for information about clusters

2. Stop customer databases.

If your enterprise contains customer databases, you can stop them using several methods, including SQL*Plus and Oracle Enterprise Manager Console.

See Also: Oracle database administration guides and *Oracle Enterprise Manager Administrator's Guide* for more information about starting databases

3. Stop the infrastructure.

If your enterprise contains more than one infrastructure, stop the primary infrastructure last.

Dependencies for Starting and Stopping Pieces of an Application Server Enterprise

You can start and stop individual pieces of an application server enterprise, for example, to reboot a system; however you must be aware of the dependencies between pieces as described in [Table 5–1](#).

Table 5–1 Dependencies for Starting and Stopping Pieces of an Application Server

Piece	Prerequisites for Starting	Prerequisites for Stopping
Application Server Instance	Associated infrastructure, if any, must be started. Customer databases used by components within the application server instance must be started.	None.
Infrastructure	None.	Associated application server instances must be stopped.
Customer Database	None	Components that use the customer database must be stopped.

Starting and Stopping Application Server Instances

This section describes how to start and stop application server instances using the Enterprise Manager Web site or command-line tools.

About the Enterprise Manager Web Site and Intelligent Agent

When starting and stopping application server instances, you should be aware of two host-wide processes:

- [Enterprise Manager Web Site \(Required\)](#)
- [Intelligent Agent \(Optional\)](#)

Enterprise Manager Web Site (Required)

The first step before starting an application server instance is to ensure that the Enterprise Manager Web site is running on the host. The Web site provides underlying processes required to run an application server instance and must be running even if you intend to use command-line tools to start your instance.

Note: Always ensure that the Enterprise Manager Web site is running on a host before starting an application server instance on that host. You must start the Enterprise Manager Web site after every system boot.

There is one Enterprise Manager Web site per host. It resides in the primary installation (or first installation) on that host. The primary installation can be an application server installation or an infrastructure. This Web site usually listens on port 1810 and provides services to all application server instances and infrastructures on that host.

To verify the status of the Enterprise Manager Web site, run the following command in the Oracle home of the primary installation:

```
(UNIX) ORACLE_HOME/bin/emctl status  
(Windows) ORACLE_HOME\bin\emctl status
```

To start the Enterprise Manager Web site, run the following command in the Oracle home of the primary installation:

```
(UNIX) ORACLE_HOME/bin/emctl start  
(Windows) ORACLE_HOME\bin\emctl start
```

You do not need to stop the Enterprise Manager Web site every time you stop an application server instance; however, if you need to stop the Web site, for example, to shut down the system, use the following commands:

```
(UNIX) ORACLE_HOME/bin/emctl stop  
(Windows) ORACLE_HOME\bin\emctl stop
```

Intelligent Agent (Optional)

You only need to run the Intelligent Agent if you are using Oracle Management Server in your enterprise. In order for Oracle Management Server to detect application server installations on a host, you must make sure the Intelligent Agent is started. Note that one Intelligent Agent is started per host and must be started after every system boot.

(UNIX) You can run the following commands in the Oracle home of the primary installation (the first installation on the host) to get status and start the Intelligent Agent:

```
ORACLE_HOME/bin/agentctl status agent  
ORACLE_HOME/bin/agentctl start agent
```

(Windows) You can check the status and start the Intelligent Agent using the Services control panel. The name of the service is in the following format:

Oracle~~ORACLE_HOME~~Agent

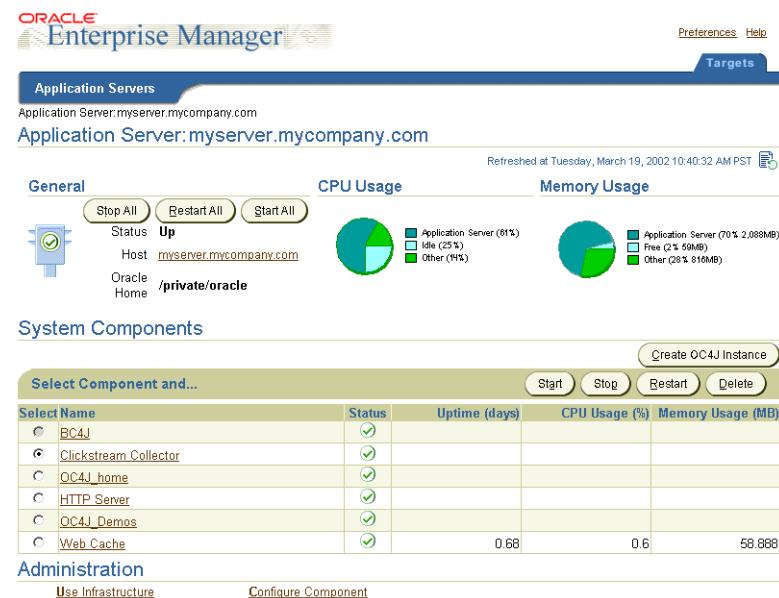
start the Intelligent Agent in the Oracle home of the primary installation:

```
ORACLE_HOME/bin/agentctl start agent
```

Starting and Stopping Instances Using the Enterprise Manager Web Site

You can start, stop, and restart all types of application server instances using the Instance Home Page on the Enterprise Manager Web site. [Figure 5–1](#) displays a sample Instance Home Page.

Figure 5–1 Instance Home Page



Starting, stopping, and restarting an application server instance applies the chosen action to the components within the instance, as described in [Table 5–2](#).

Table 5–2 Application Server Instance Start/Stop/Restart Actions

Performing this action on an application server instance:	Applies these actions to its components:
Start	Starts components that are stopped. Does not affect components that are already running.
Stop	Stops components that are running. Does not affect components that are stopped.
Restart	Restarts components that are running. Starts components that are stopped.

To start, stop, or restart an application server instance:

1. Navigate to the Instance Home Page on the Enterprise Manager Web site.
2. In the General section of the Instance Home Page, click **Start All**, **Stop All**, or **Restart All**. In the System Components section, review the status of each component and confirm it was started or stopped.

See Also: If the application server instance is part of a cluster, refer to "[Managing the Cluster](#)" on page 14-28 for information on starting and stopping an instance as part of a cluster.

Starting and Stopping Instances Using Commands

You can start and stop a J2EE and Web Cache instance using command-line tools. J2EE and Web Cache is the only install type that has command-line tools for starting and stopping all of its components. You must start and stop all other install types using the Enterprise Manager Web site.

Starting a J2EE and Web Cache Instance Using Commands

To start a J2EE and Web Cache instance using command-line tools:

1. Start the Enterprise Manager Web site. Even though you are not using the Web site, this ensures that the processes to support the `dcmctl` command-line tool are started. To start the Web site, execute the following command in the Oracle home of the primary installation on your host:

```
(UNIX) ORACLE_HOME/bin/emctl start  
(Windows) ORACLE_HOME\bin\emctl start
```

2. Start Oracle HTTP Server and OC4J (the rest of the commands in this section should be executed in the Oracle home of the J2EE and Web Cache instance):

```
(UNIX) ORACLE_HOME/dcm/bin/dcmctl start  
(Windows) ORACLE_HOME\dcm\bin\dcmctl start
```

3. If Web Cache is configured, start Web Cache:

```
(UNIX) ORACLE_HOME/bin/webcachectl start  
(Windows) ORACLE_HOME\bin\webcachectl start
```

Stopping a J2EE and Web Cache Instance Using Commands

To stop a J2EE and Web Cache instance using command-line tools:

1. If Web Cache is configured, stop Web Cache:

(UNIX) *ORACLE_HOME/bin/webcachectl stop*
(Windows) *ORACLE_HOME\bin\webcachectl stop*

2. Stop Oracle HTTP Server and OC4J:

(UNIX) *ORACLE_HOME/dcm/bin/dcmctl stop*
(Windows) *ORACLE_HOME\dcm\bin\dcmctl stop*

Starting and Stopping Components

You can start, stop, and restart individual components using the Instance Home Page or the component home page on the Enterprise Manager Web site. You can also start and stop some components using command-line tools.

Warning: Starting or stopping an individual component can affect the behavior of other components in the application server instance. Before starting or stopping a component, review the information in "[Dependencies for Starting and Stopping Components](#)" on page 5-13.

Starting and Stopping Components Using the Instance Home Page

To start, stop, or restart a component from the Instance Home Page:

1. Navigate to the Instance Home Page on the Enterprise Manager Web site. Scroll to the System Components section.
2. Click the radio button in the Select column for the component.
3. Click the **Start**, **Stop**, or **Restart** button on the top right of the System Components section.

See Also: The radio button in the Select column is disabled for some components. See "[About Components that Cannot Be Started or Stopped](#)" on page 5-12 for more information.

Starting and Stopping Components Using the Component Home Page

To start, stop, or restart a component from the home page for the component:

1. Navigate to the Instance Home Page on the Enterprise Manager Web site. Scroll to the System Components section.
2. Select the component in the Name column. This opens the home page for the component.
3. On the component home page, click the **Start**, **Stop**, or **Restart** button in the General section.

See Also: The home page for some components does not allow you to start, stop, and restart the component. See "[About Components that Cannot Be Started or Stopped](#)" on page 5-12 for more information.

Starting and Stopping Components Using Commands

You can start and stop some components using command-line tools. [Table 5-3](#) lists the commands for UNIX and [Table 5-4](#) lists the commands for Windows.

Note that many components are started when their corresponding OC4J instance is started. Refer to "[About Components that Cannot Be Started or Stopped](#)" on page 5-12.

Table 5-3 Starting and Stopping Components Using Command-Line Tools on UNIX

Component	Action	Command
Oracle HTTP Server	Start	<i>ORACLE_HOME/dcm/bin/dcmctl start -ct ohs</i>
	Stop	<i>ORACLE_HOME/dcm/bin/dcmctl stop -ct ohs</i>
Individual OC4J Instances	Start	<i>ORACLE_HOME/dcm/bin/dcmctl start -co instance_name</i>
	Stop	<i>ORACLE_HOME/dcm/bin/dcmctl stop -co instance_name</i>
All OC4J Instances	Start	<i>ORACLE_HOME/dcm/bin/dcmctl start -ct oc4j</i>
	Stop	<i>ORACLE_HOME/dcm/bin/dcmctl stop -ct oc4j</i>
Web Cache	Start	<i>ORACLE_HOME/bin/webcachectl start</i>
	Stop	<i>ORACLE_HOME/bin/webcachectl stop</i>
Discoverer	Start	<i>ORACLE_HOME/discoverer902/util/startall.sh</i>
	Stop	<i>ORACLE_HOME/discoverer902/util/stopall.sh</i>
Reports	Start	<i>ORACLE_HOME/bin/rwserver.sh server=name</i>
	Stop	<i>ORACLE_HOME/bin/rwserver.sh server=name shutdown=yes</i>

Table 5–4 Starting and Stopping Components Using Command-Line Tools on Windows

Component	Action	Command
Oracle HTTP Server	Start	<i>ORACLE_HOME\dcm\bin\dcmctl start -ct ohs</i>
	Stop	<i>ORACLE_HOME\dcm\bin\dcmctl stop -ct ohs</i>
Individual OC4J Instances	Start	<i>ORACLE_HOME\dcm\bin\dcmctl start -co instance_name</i>
	Stop	<i>ORACLE_HOME\dcm\bin\dcmctl stop -co instance_name</i>
All OC4J Instances	Start	<i>ORACLE_HOME\dcm\bin\dcmctl start -ct oc4j</i>
	Stop	<i>ORACLE_HOME\dcm\bin\dcmctl stop -ct oc4j</i>
Web Cache	Start	<i>ORACLE_HOME\bin\webcachectl start</i>
	Stop	<i>ORACLE_HOME\bin\webcachectl stop</i>
Reports	Start	<i>ORACLE_HOME\bin\rwserver server=name</i>
	Stop	<i>ORACLE_HOME\bin\rwserver server=name shutdown=yes</i>

About Components that Cannot Be Started or Stopped

You cannot start or stop some components. The radio buttons in the Select column on the Instance Home Page are disabled for these components, and their component home pages do not have Start, Stop, or Restart buttons.

This section provides information about the following components, which cannot be started or stopped:

- [BC4J](#)
- [Discoverer](#)
- [Forms](#)
- [Internet File System](#)
- [JServ](#)
- [Portal](#)
- [Single Sign-On](#)

BC4J

Business Components for Java (BC4J) is a framework for building applications and is deployed to the OC4J_home instance. BC4J displays an up status if the OC4J_home instance is running and a down status if the OC4J_home instance is stopped.

Discoverer

Discoverer is provided by a servlet running in the OC4J_BI_Forms instance. Discoverer displays an up status if the OC4J_BI_Forms instance is running and a down status if the OC4J_BI_Forms instance is stopped.

Forms

The Forms service is provided by a servlet running in the OC4J_BI_Forms instance. Forms displays an up status if the OC4J_BI_Forms instance is running and a down status if the OC4J_BI_Forms instance is stopped.

Internet File System

Internet File System is supported by a servlet running in the OC4J_iFS instance. Internet File System displays an up status if the OC4J_iFS instance is running and a down status if the OC4J_iFS instance is stopped.

JServ

JServ is a Java servlet environment available in the HTTP Server. Jserv displays an up status if HTTP Server is running and a down status if HTTP Server is stopped.

Portal

Portal is supported by a servlet running in the OC4J_Portal instance. Portal displays an up status if the OC4J_Portal instance is running and a down status if the OC4J_Portal instance is stopped.

Single Sign-On

Single Sign-On service is accessible through HTTP Server. Single Sign-On displays an up status if HTTP Server is running and a down status if HTTP Server is stopped.

Dependencies for Starting and Stopping Components

There are many dependencies between components in an application server instance. Before you start or stop an individual component, you must understand the effect, if any, the action will have on other components.

You may need to make sure other components are started before starting a component. Stopping a component may disrupt the services of other components.

[Table 5–5](#), describes the prerequisites for starting and the side-effects of stopping individual components.

Table 5–5 Dependencies for Starting and Stopping Components

Component	Prerequisites for Starting	Side-Effects of Stopping
Clickstream Collector	None	None
Clickstream	None	None
HTTP Server	None	All services that require access through the HTTP Server will be unavailable.
Internet Directory	Metadata Repository must be started.	All application server instances using the directory will be disrupted. The directory should only be stopped as part of an orderly infrastructure shutdown.
OC4J_BI_Forms	None	Forms, Reports, Clickstream, Personalization, and Discoverer become unavailable.
OC4J_DAS	HTTP Server and OC4J_home must be started.	Delegated Administration Service becomes unavailable.
OC4J_Demos	None	Demos that depend on OC4J_Demos will not work.
OC4J_home (OC4J default instance)	None	None
OC4J_iFS	None	Internet File System becomes unavailable.
OC4J_Portal	None	Portal becomes unavailable.
OC4J_UM	None	Unified Messaging becomes unavailable.
OC4J_Wireless	None	None.
Reports	None	None
Unified Messaging	HTTP Server and OC4J_home must be started.	None <i>Note: Restarting causes Unified Messaging to reload operational settings. It does not stop and start the instance.</i>
Web Cache	None	Portal will become unavailable.
Wireless	None	None

Starting an Infrastructure

This section contains the following procedures for starting an infrastructure:

- [Starting an Infrastructure Using Oracle Enterprise Manager on UNIX and Windows](#)
- [Starting an Infrastructure Using Commands on UNIX](#)
- [Starting an Infrastructure Using Commands on Windows](#)

No matter which procedure you use, starting an infrastructure involves performing the following steps in order:

1. Start the Metadata Repository
2. Start Oracle Internet Directory
3. Start the Enterprise Manager Web site.
4. Start Oracle HTTP Server
5. Start the OC4J_DAS instance
6. Start Web Cache (optional)
7. Start Oracle Management Server and Intelligent Agent (optional)

About Unused Component Instances in the Infrastructure

An infrastructure contains several component instances that are not required to run an infrastructure and incur unnecessary system overhead if started. When you start an infrastructure, you do not need to start these components:

- OC4J_Demos: This OC4J instance is only required if you want to run demos.
- OC4J_home: This OC4J instance is not required because applications are not deployed to an infrastructure.
- BC4J: This component is only used by OC4J_home and is not required if OC4J_home is not started.
- Clickstream Collector: This instance is only required to collect Clickstream data and is not necessary since applications are not deployed to an infrastructure.

[Figure 5–2](#) shows the component statuses of a properly started infrastructure.

Figure 5–2 A Properly Started Infrastructure Instance

System Components

The screenshot shows a web-based interface for managing Oracle Application Server components. At the top right are four buttons: 'Create OC4J Instance' (with a '+' icon), 'Start', 'Stop', 'Restart', and 'Delete'. Below this is a table titled 'System Components' with the following data:

Select Name	Status	Uptime (days)	CPU Usage (%)	Memory Usage (MB)
<input type="radio"/> BC4J	!			
<input checked="" type="radio"/> Clickstream Collector	!			
<input type="radio"/> HTTP Server	✓	0.05	0.2	105.117
<input type="radio"/> Internet Directory	✓	3.999	0	16.273
<input type="radio"/> OC4J_DAS	✓	0.049	0.1	58.414
<input type="radio"/> OC4J_Demos	!			
<input type="radio"/> OC4J_home	!			
<input type="radio"/> Single Sign-On:orasso:7777	✓			

Starting an Infrastructure Using Oracle Enterprise Manager on UNIX and Windows

This section describes how to start an infrastructure using Oracle Enterprise Manager on UNIX and Windows.

1. Start the metadata repository.

- Start the metadata repository listener:

```
(UNIX) ORACLE_HOME/bin/lsnrctl start  
(Windows) ORACLE_HOME\bin\lsnrctl start
```

- Launch Oracle Enterprise Manager in standalone mode:

```
(UNIX) ORACLE_HOME/bin/oemapp dbastudio  
(Windows) ORACLE_HOME\bin\oemapp dbastudio
```

Select **Launch Standalone** and click **OK**.

- Expand the Databases entry in the System Objects frame by clicking the plus sign (+) next to it.
- Select the metadata repository entry.
- In the toolbar, select Object -> Connect. Connect as a user with SYSDBA privileges.
- In the toolbar, select Object -> Startup.

2. Start Oracle Internet Directory.

- Set the ORACLE_SID is set to the metadata repository system identifier (default is iasdb).
- Start the Oracle Internet Directory monitor:

```
(UNIX) ORACLE_HOME/bin/oidmon start  
(Windows) ORACLE_HOME\bin\oidmon start
```

- Start the Oracle Internet Directory server:

```
(UNIX) ORACLE_HOME/bin/oidctl server=oidldapd configset=0 instance=n start  
(Windows) ORACLE_HOME\bin\oidctl server=oidldapd configset=0 instance=n start
```

where *n* is any instance number (1, 2, 3...) that is not in use. For example:

```
(UNIX) ORACLE_HOME/bin/oidctl server=oidldapd configset=0 instance=1 start
```

```
(Windows) ORACLE_HOME\bin\oidctl server=oidldapd configset=0 instance=1 start
```

3. **Start the Enterprise Manager Web site.** The Web site must be started after every system boot.

(UNIX) You can run the following commands in the Oracle home of the primary installation (the first installation on the host) to get status and start the Web site:

```
ORACLE_HOME/bin/emctl status  
ORACLE_HOME/bin/emctl start
```

(Windows) You can check the status and start the Enterprise Manager Web site using the Services control panel. The name of the service is in the following format:

OracleORACLE_HOMEEMwebsite

4. **Access the Infrastructure Instance Page on the Enterprise Manager Web site.**

Point your browser to the Enterprise Manager Web site on your host using the following URL:

<http://hostname:1810>

Log in as `ias_admin`. This opens the Farm Page. Select the infrastructure instance in the Name column in the Standalone Instances section. This opens the Infrastructure Instance Page.

Note that you should **not** click **Start All**. This will start OC4J instances (OC4J_home and OC4J_Demos) that are not required and incur unnecessary overhead. Instead, follow the rest of these steps to start only the required components. Refer to "[About Unused Component Instances in the Infrastructure](#)" on page 5-15 for more information.

5. **Start Oracle HTTP Server.**

- a. On the Infrastructure Instance Page, in the System Components section, click the radio button in the Select column for HTTP Server.
- b. Click the **Start** button on the top right of the System Components section.

Note that starting Oracle HTTP Server also makes Oracle9iAS Single Sign-On available.

6. Start the OC4J_DAS instance.

- a. On the Infrastructure Instance Page, in the System Components section, click the radio button in the Select column for OC4J_DAS.
- b. Click the **Start** button on the top right of the System Components section.

Note that the infrastructure instance contains other OC4J instances, such as OC4J_home and OC4J_Demos, but these do not need to be started; their services are not required and incur unnecessary overhead.

7. Start Web Cache (optional).

Web Cache is not configured in the infrastructure by default, but if you have configured it, start it as follows:

- a. On the Infrastructure Instance Page, in the System Components section, click the radio button in the Select column for Web Cache.
- b. Click the **Start** button on the top right of the System Components section.

8. Start Oracle Management Server and Intelligent Agent (optional).

Perform these steps only if you have configured Oracle Management Server.

- a. Start Oracle Management Server:

(UNIX) `ORACLE_HOME/bin/oemctl start oms`
(Windows) `ORACLE_HOME\bin\oemctl start oms`

- b. Start the Intelligent Agent.

In order for Oracle Management Server to detect the infrastructure and any other application server installations on this host, you must make sure the Intelligent Agent is started. Note that one Intelligent Agent is started per host and must be started after every reboot.

(UNIX) You can run the following commands in the Oracle home of the primary installation (the first installation on the host) to get status and start the Intelligent Agent:

`ORACLE_HOME/bin/agentctl status agent`
`ORACLE_HOME/bin/agentctl start agent`

(Windows) You can check the status and start the Intelligent Agent using the Services control panel. The name of the service is in the following format:

Oracle`ORACLE_HOME`Agent

Starting an Infrastructure Using Commands on UNIX

This section describes how to start an infrastructure using command-line tools on UNIX. Except where noted, all commands should be run in the Oracle home of the infrastructure.

1. Start the metadata repository.

- a. Start the metadata repository listener:

```
ORACLE_HOME/bin/lsnrctl start
```

- b. Set the ORACLE_SID environment variable to the metadata repository system identifier. The default ORACLE_SID is iasdb.

```
(Bourne or Korn shell) ORACLE_SID=iasdb; export ORACLE_SID  
(C shell) setenv ORACLE_SID iasdb
```

- c. Start the metadata repository instance using SQL*Plus:

```
ORACLE_HOME/bin/sqlplus /nolog  
sql> connect sys/password_for_sys as sysdba  
sql> startup  
sql> quit
```

2. Start Oracle Internet Directory.

- a. Make sure the ORACLE_SID is set to the metadata repository system identifier (refer to previous step).

- b. Start the Oracle Internet Directory monitor:

```
ORACLE_HOME/bin/oidmon start
```

- c. Start the Oracle Internet Directory server:

```
ORACLE_HOME/bin/oidctl server=oidldapd configset=0 instance=n start
```

where *n* is any instance number (1, 2, 3...) that is not in use. For example:

```
ORACLE_HOME/bin/oidctl server=oidldapd configset=0 instance=1 start
```

3. Start the Enterprise Manager Web site.

Even though you are using command-line, the Web site is required because it provides underlying support for the command-line tools. The Web site must be started after every system boot.

You can run the following commands in the Oracle home of the primary installation (the first installation on the host) to get status and start the Web site:

```
ORACLE_HOME/bin/emctl status  
ORACLE_HOME/bin/emctl start
```

4. Start Oracle HTTP Server.

```
ORACLE_HOME/dcm/bin/dcmctl start -ct ohs
```

Note that starting Oracle HTTP Server also makes Oracle9iAS Single Sign-On available.

5. Start the OC4J_DAS instance.

```
ORACLE_HOME/dcm/bin/dcmctl start -co OC4J_DAS
```

Note that the infrastructure instance contains other OC4J instances, such as OC4J_home and OC4J_Demos, but these do not need to be started; their services are not required and incur unnecessary overhead. Refer to "[About Unused Component Instances in the Infrastructure](#)" on page 5-15 for more information.

6. Start Web Cache (optional).

Web Cache is not configured in the infrastructure by default, but if you have configured it, start it as follows:

```
ORACLE_HOME/bin/webcachectl start
```

7. Start Oracle Management Server and Intelligent Agent (optional).

Perform these steps only if you have configured Oracle Management Server.

a. Start Oracle Management Server:

```
ORACLE_HOME/bin/oemctl start oms
```

b. Start the Intelligent Agent.

In order for Oracle Management Server to detect the infrastructure and any other application server installations on this host, you must make sure the Intelligent Agent is started. Note that one Intelligent Agent is started per host and must be started after every reboot.

You can run the following commands in the Oracle home of the primary installation (the first installation on the host) to get status and start the Intelligent Agent:

```
ORACLE_HOME/bin/agentctl status agent  
ORACLE_HOME/bin/agentctl start agent
```

Starting an Infrastructure Using Commands on Windows

This section describes how to start an infrastructure using command-line tools on Windows. Except where noted, all commands should be run in the Oracle home of the infrastructure.

1. Start the metadata repository.

- a. Start the metadata repository listener:

```
ORACLE_HOME\bin\lsnrctl start
```

- b. Set the ORACLE_SID environment variable to the metadata repository system identifier (default is `iasdb`).

You can set the ORACLE_SID system variable using the System Properties control panel.

- c. Start the metadata repository instance using SQL*Plus:

```
ORACLE_HOME\bin\sqlplus /nolog  
sql> connect sys/password_for_sys as sysdba  
sql> startup  
sql> quit
```

2. Start Oracle Internet Directory.

- a. Make sure the ORACLE_SID is set to the metadata repository system identifier (refer to previous step).

- b. Start the Oracle Internet Directory monitor:

```
ORACLE_HOME\bin\oidmon start
```

- c. Start the Oracle Internet Directory server:

```
ORACLE_HOME\bin\oidctl server=oidldapd configset=0 instance=n start
```

where *n* is any instance number (1, 2, 3...) that is not in use. For example:

```
ORACLE_HOME\bin\oidctl server=oidldapd configset=0 instance=1 start
```

3. Start the Enterprise Manager Web site.

Even though you are using command-line, the Web site is required because it provides underlying support for the command-line tools. The Web site must be started after every system boot.

You can check the status and start the Enterprise Manager Web site using the Services control panel. The name of the service is in the following format:

OracleORACLE_HOMEEMwebsite

You can also start the service using the following command line:

```
net start WEB_SITE_SERVICE_NAME
```

4. Start Oracle HTTP Server.

```
ORACLE_HOME\dcm\bin\dcmctl start -ct ohs
```

Note that starting Oracle HTTP Server also makes Oracle9iAS Single Sign-On available.

5. Start the OC4J_DAS instance.

```
ORACLE_HOME\dcm\bin\dcmctl start -co OC4J_DAS
```

Note that the infrastructure instance contains other OC4J instances, such as OC4J_home and OC4J_Demos, but these do not need to be started; their services are not required and incur unnecessary overhead. Refer to "[About Unused Component Instances in the Infrastructure](#)" on page 5-15 for more information.

6. Start Web Cache (optional).

Web Cache is not configured in the infrastructure by default, but if you have configured it, start it as follows:

```
ORACLE_HOME\bin\webcachectl start
```

7. Start Oracle Management Server and Intelligent Agent (optional).

Perform these steps only if you have configured Oracle Management Server.

a. Start Oracle Management Server:

```
ORACLE_HOME\bin\oemctl start oms
```

b. Start the Intelligent Agent.

In order for Oracle Management Server to detect the infrastructure and any other application server installations on this host, you must make sure the Intelligent Agent is started. Note that one Intelligent Agent is started per host and must be started after every reboot.

You can check the status and start the Intelligent Agent using the Services control panel. The name of the service is in the following format:

OracleORACLE_HOMEAgent

Stopping an Infrastructure

This section contains the following procedures for stopping an infrastructure:

- [Stopping an Infrastructure Using Oracle Enterprise Manager on UNIX and Windows](#)
- [Stopping an Infrastructure Using Commands on UNIX](#)
- [Stopping an Infrastructure Using Commands on Windows](#)

No matter which procedure you use, stopping an infrastructure involves performing the following steps in order:

1. Stop all middle-tier application server instances that use the infrastructure.
2. Stop Oracle Management Server and Intelligent Agent (optional)
3. Stop Web Cache (optional)
4. Stop OC4J instances
5. Stop Oracle HTTP Server
6. Stop Oracle Internet Directory
7. Stop the Metadata Repository

Stopping an Infrastructure Using Oracle Enterprise Manager on UNIX and Windows

This section describes how to stop an infrastructure using Oracle Enterprise Manager on UNIX and Windows.

- 1. Stop all middle-tier application server instances that use the infrastructure.**
- 2. Stop Oracle Management Server and Intelligent Agent (optional).**

Perform these steps only if you have configured Oracle Management Server.

- a. Stop Oracle Management Server:**

```
(UNIX) $ORACLE_HOME/bin/oemctl stop oms  
(Windows) %ORACLE_HOME%\bin\oemctl stop oms
```

- b. Stop the Intelligent Agent (for system shutdown only).**

If you are preparing to shut down your system, stop the Intelligent Agent on this host.

(UNIX) You can run the following commands in the Oracle home of the primary installation (the first installation on the host) to get status and stop the Intelligent Agent:

```
$ORACLE_HOME/bin/agentctl status agent  
$ORACLE_HOME/bin/agentctl stop agent
```

(Windows) You can check the status and start the Intelligent Agent using the Services control panel. The name of the service is in the following format:

Oracle\$ORACLE_HOMEAgent

- 3. Access the Infrastructure Instance Page on the Enterprise Manager Web site.**

Point your browser to the Enterprise Manager Web site on your host using the following URL:

<http://hostname:1810>

Log in as `ias_admin`. This opens the Farm Page. Select the infrastructure instance in the Name column in the Standalone Instances section. This opens the Infrastructure Instance Page.

Note that you should **not click Stop All**. Instead, follow the rest of these steps to stop components in the proper order.

4. Stop Web Cache (optional).

Web Cache is not configured in the infrastructure by default, but if you have configured it, stop it as follows:

- a. On the Infrastructure Instance Page, in the System Components section, click the radio button in the Select column for Web Cache.
- b. Click the **Stop** button on the top right of the System Components section.

5. Stop the OC4J instances.

Stop all running OC4J instances (OC4J_DAS, OC4J_home, OC4J_Demos) by performing these steps on each one.

- a. On the Infrastructure Instance Page, in the System Components section, click the radio button in the Select column for the OC4J instance.
- b. Click the **Stop** button on the top right of the System Components section.

6. Stop Oracle HTTP Server.

- a. On the Infrastructure Instance Page, in the System Components section, click the radio button in the Select column for HTTP Server.
- b. Click the **Stop** button on the top right of the System Components section.

Note that stopping Oracle HTTP Server also makes Oracle9iAS Single Sign-On unavailable.

7. Stop Oracle Internet Directory.

- a. On the Infrastructure Instance Page, in the System Components section, click the radio button in the Select column for Internet Directory.
- b. Click the **Stop** button on the top right of the System Components section.

8. Stop the metadata repository.

- a. Launch Oracle Enterprise Manager in standalone mode:

(UNIX) *ORACLE_HOME/bin/oemapp dbastudio*
(Windows) *ORACLE_HOME\bin\oemapp dbastudio*

Select **Launch Standalone** and click **OK**.

- b. Expand the Databases entry in the System Objects frame by clicking the plus sign (+) next to it.
- c. Select the metadata repository entry.

- d. In the toolbar, select Object -> Connect. Connect as a user with SYSDBA privileges.
- e. In the toolbar, select Object -> Shutdown.
- f. Stop the metadata repository listener:

(UNIX) *ORACLE_HOME/bin/lsnrctl stop*
(Windows) *ORACLE_HOME\bin\lsnrctl stop*

9. Stop the Enterprise Manager Web site (system shutdown only).

If you are preparing to shut down your system, stop the Enterprise Manager Web site.

(UNIX) You can run the following commands in the Oracle home of the primary installation (the first installation on the host) to get status and stop the Web site:

ORACLE_HOME/bin/emctl status
ORACLE_HOME/bin/emctl stop

(Windows) You can check the status and stop the Enterprise Manager Web site using the Services control panel. The name of the service is in the following format:

Oracle*ORACLE_HOME*EMwebsite

Stopping an Infrastructure Using Commands on UNIX

This section describes how to stop an infrastructure using command-line tools on UNIX. Except where noted, all commands should be run in the Oracle home of the infrastructure.

- 1. Stop all middle-tier application server instances that use the infrastructure.**
- 2. Stop Oracle Management Server and Intelligent Agent (optional).**

Perform these steps only if you have configured Oracle Management Server.

- a. Stop Oracle Management Server:**

If you have configured Oracle Management Server, stop it as follows:

```
ORACLE_HOME/bin/oemctl stop oms
```

- b. Stop the Intelligent Agent (for system shutdown only).**

If you are preparing to shut down your system, stop the Intelligent Agent on this host.

You can run the following commands in the Oracle home of the primary installation (the first installation on the host) to get status and stop the Intelligent Agent:

```
ORACLE_HOME/bin/agentctl status agent  
ORACLE_HOME/bin/agentctl stop agent
```

- 3. Stop the Enterprise Manager Web site (for system shutdown only).**

If you are preparing to shut down your system, stop the Enterprise Manager Web site by running the following command in the Oracle home of the primary installation (the first installation on the host):

```
ORACLE_HOME/bin/emctl stop
```

- 4. Stop Web Cache (optional).**

Web Cache is not configured in the infrastructure by default, but if you have configured it, stop it as follows:

```
ORACLE_HOME/bin/webcachectl stop
```

- 5. Stop OC4J instances and Oracle HTTP Server.**

If you are preparing to shutdown your system, stop all running OC4J instances, Oracle HTTP Server, and OPMN as follows:

```
ORACLE_HOME/dcm/bin/dcmctl shutdown
```

Otherwise, if you are not preparing to shut down your system, stop all running OC4J instances and Oracle HTTP Server (leave OPMN running):

```
ORACLE_HOME/dcm/bin/dcmctl stop
```

Note that stopping Oracle HTTP Server also makes Oracle9iAS Single Sign-On unavailable.

6. Stop Oracle Internet Directory.

- a. Make sure the ORACLE_SID is set to the metadata repository system identifier (refer to the next step).
- b. Stop the Oracle Internet Directory server:

```
ORACLE_HOME/bin/oidctl server=oidldapd configset=0 instance=n stop
```

where *n* is the instance number that was used to start the server (1, 2, 3...).
For example:

```
ORACLE_HOME\bin\oidctl server=oidldapd configset=0 instance=1 stop
```

- c. Stop the Oracle Internet Directory monitor:

```
ORACLE_HOME/bin/oidmon stop
```

7. Stop the metadata repository.

- a. Set the ORACLE_SID environment variable to the metadata repository system identifier. The default ORACLE_SID is iasdb.

(Bourne or Korn shell) ORACLE_SID=iasdb; export ORACLE_SID
(C shell) setenv ORACLE_SID iasdb

- b. Stop the metadata repository instance using SQL*Plus:

```
ORACLE_HOME/bin/sqlplus /nolog
sql> connect sys/password_for_sys as sysdba
sql> shutdown
sql> quit
```

- c. Stop the metadata repository listener:

```
ORACLE_HOME/bin/lsmrctl stop
```

Stopping an Infrastructure Using Commands on Windows

This section describes how to stop an infrastructure using command-line tools on Windows. Except where noted, all commands should be run in the Oracle home of the infrastructure.

- 1. Stop all middle-tier application server instances that use the infrastructure.**
- 2. Stop Oracle Management Server and Intelligent Agent (optional).**

Perform these steps only if you have configured Oracle Management Server.

- a. Stop Oracle Management Server.**

```
ORACLE_HOME\bin\oemctl stop oms
```

- b. Stop the Intelligent Agent (for system shutdown only).**

If you are preparing to shut down your system, stop the Intelligent Agent on this host.

The name of the service is in the following format:

```
OracleORACLE_HOMEAgent
```

For example:

```
OracleOraHome902Agent
```

The Services control panel shows you the status of the service and indicates the method used to stop the service. You can then select the service name and use the controls on the Services window to stop the service.

- 3. Stop the Enterprise Manager Web site (for system shutdown only).**

If you are preparing to shut down your system, stop the Enterprise Manager Web site.

You can check the status and stop the Enterprise Manager Web site using the Services control panel. The name of the service is in the following format:

```
OracleORACLE_HOMEEMwebsite
```

For example, if the name of the Oracle home is OraHome902, the service name is:

```
OracleOraHome902EMWebsite
```

4. Stop Web Cache (optional).

Web Cache is not configured in the infrastructure by default, but if you have configured it, stop it as follows:

```
ORACLE_HOME\bin\webcachectl stop
```

5. Stop OC4J instances and Oracle HTTP Server

If you are preparing to shutdown your system, stop all running OC4J instances, Oracle HTTP Server, and OPMN as follows:

```
ORACLE_HOME\dcm\bin\dcmctl shutdown
```

Otherwise, if you are not preparing to shut down your system, stop all running OC4J instances and Oracle HTTP Server (leave OPMN running):

```
ORACLE_HOME\dcm\bin\dcmctl stop
```

Note that stopping Oracle HTTP Server also makes Oracle9iAS Single Sign-On unavailable.

6. Stop Oracle Internet Directory.

a. Make sure the ORACLE_SID is set to the metadata repository system identifier (refer to next step).

b. Stop the Oracle Internet Directory server:

```
ORACLE_HOME\bin\oidctl server=oidldapd configset=0 instance=n stop
```

where *n* is the instance number that was used to start the server (1, 2, 3...).
For example:

```
ORACLE_HOME\bin\oidctl server=oidldapd configset=0 instance=1 stop
```

c. Stop the Oracle Internet Directory monitor:

```
ORACLE_HOME\bin\oidmon stop
```

7. Stop the metadata repository.

- a.** Set the ORACLE_SID environment variable to the metadata repository system identifier. The default ORACLE_SID is iasdb.

You can set the ORACLE_SID system variable using the System Properties control panel.

- b.** Stop the metadata repository instance using SQL*Plus:

```
ORACLE_HOME\bin\sqlplus /nolog  
sql> connect sys/password_for_sys as sysdba  
sql> shutdown  
sql> quit
```

- c.** Stop the metadata repository listener:

```
ORACLE_HOME\bin\lsnrctl stop
```

6

Configuring HTTP Server, OC4J, and Web Cache

This chapter provides an introduction to configuring these key components: Oracle HTTP Server, OC4J, and Oracle9iAS Web Cache.

It contains the following topics:

- [Introduction](#)
- [About Manually Editing HTTP Server and OC4J Configuration Files](#)
- [Configuring Oracle HTTP Server](#)
- [Enabling JServ](#)
- [Configuring Oracle9iAS Containers for J2EE](#)
- [Configuring Oracle9iAS Web Cache](#)

Introduction

This chapter provides an introduction to configuring components using the Oracle Enterprise Manager Web site. It discusses common configuration tasks associated with three key components: Oracle HTTP Server, Oracle9iAS Containers for J2EE, and Oracle9iAS Web Cache.

See Also: Enterprise Manager Web site allows you to configure many other Oracle9iAS components. For more information, refer to the individual component guides and use the Enterprise Manager Web site online help.

About Manually Editing HTTP Server and OC4J Configuration Files

If you edit Oracle HTTP Server or OC4J configuration files manually, instead of using the Enterprise Manager Web site, you must use the DCM command-line utility `dcmctl` to notify the DCM repository of the changes. Otherwise, your changes will not go into effect and will not be reflected in the Enterprise Manager Web site. [Table 6-1](#) lists the `dcmctl` commands. Note that the `dcmctl` tool is located in:

(UNIX) `ORACLE_HOME/dcm/bin/dcmctl`
(Windows) `ORACLE_HOME\dcm\bin\dcmctl`

Table 6-1 dcmctl Commands for Updating the DCM Repository

To notify DCM of changes made to:	Use this command:
Oracle HTTP Server configuration files	<code>dcmctl updateConfig -ct ohs</code>
OC4J configuration files	<code>dcmctl updateConfig -ct oc4j</code>
All configuration files	<code>dcmctl updateConfig</code>

Before you change configuration parameters, manually or using the Enterprise Manager Web site, you can save the current state of your configuration files and installed J2EE applications with the following command:

```
dcmctl saveInstance -dir directory_name
```

You can then restore the state and back out of any subsequent changes that were made using the following command:

```
dcmctl restoreInstance [ -dir directory_name ]
```

See Also: [Appendix F, "DCM Command-Line Utility \(dcmctl\)"](#) for more information

Configuring Oracle HTTP Server

You can configure Oracle HTTP Server using the Oracle HTTP Server Home Page on the Oracle Enterprise Manager Web site. You can perform tasks such as modifying directives, changing log properties, specifying a port for a listener, managing client requests, and editing server configuration files.

You can access the Oracle HTTP Server Home Page in the Name column of the System Components table on the Instance Home Page. [Figure 6–1](#) displays the Oracle HTTP Server Home Page.

This section describes the following tasks:

- [Modifying the Document Root, Administrator E-mail, User, and Group Settings](#)
- [Specifying a Port for a Listener](#)
- [Changing the Error Log Properties](#)
- [Adding an Access Log File](#)
- [Changing the Access Log Properties](#)
- [Managing the Client Request and Connection Handling](#)
- [Editing the Server Configuration Files](#)
- [Creating a Database Access Descriptor \(DAD\) for mod_plsql](#)

Figure 6–1 HTTP Server Home Page

The screenshot shows the Oracle Enterprise Manager Application Servers interface. The title bar says "Enterprise Manager". The top navigation bar includes "Targets" and "Help". Below the title bar, the URL is "Farm > Application Server: ias1.myserver.mycompany.com > HTTP Server". A status message indicates it was "Refreshed at Wednesday, March 6, 2002 2:27:59 PM PST".

General

	Status	Up	Stop	Restart
	Start	2002		
	Time	11:24:34 AM PST		

Default Server Configuration

Server Name	myserver.mycompany.com
Document Root	/private/oracle/OraHome1
Last Modification	Mar 6, 2002 11:20:32 AM PST

Status

Heap Usage (MB)	20.094
CPU Usage (%)	0.6
Memory Usage (MB)	295.752
Error Rate (%)	0
Active Connections	2
Connection Open Time (sec)	0.02

Response and Load

Active Requests	1
Request Throughput (reqs/sec)	0.53
Request Processing Time (sec)	0.01
Data Throughput (KB/sec)	8.5
Data Processed (KB/req)	16.11

See Also: If you are unfamiliar with the Oracle HTTP Server configuration settings, see the *Oracle HTTP Server Administration Guide* before making any changes.

Modifying the Document Root, Administrator E-mail, User, and Group Settings

After you start Oracle HTTP Server, the system is ready to listen for and respond to requests. You may need to make modifications to the document root, administrator e-mail, user, and group settings in order to process requests efficiently.

- The **document root** is the directory from which the server *serves* files. Note that the document root directory is different from the server root directory, which is only used to *store* the server files.
- The **administrator e-mail** address is the server's main contact. This address receives notifications if the server experiences error conditions.
- The **user** name specifies the userid to which the server answers requests. In order to use this directive, the server must be run initially as root. The user should have privileges to access files that are available for everyone, and the user should be able to execute code which is not meant for HTTP requests. It is recommended that you set up a new user for running the server.

- The **group** name specifies the group under which the server answers requests. In order to use this directive, the standalone server must be run initially as root. It is recommended that you create a new group for running the server.

To modify these settings:

1. Navigate to the Oracle HTTP Server Home Page on the Oracle Enterprise Manager Web site. Scroll to the Administration section.
2. Select **Server Properties** under the Administration section. This opens the Oracle HTTP Server Properties Page.
3. Type a new path in the **Document Root** field to change the document root directory. The path should be relative to the Server Root Directory specified during initial configuration.
4. Type the appropriate email address in the **Administrator E-Mail** field. Oracle HTTP Server uses this email address to issue notices and warnings. The administrator should have full privileges.
5. Add or change the **User** identifier by typing a new user name in the fields provided.
6. Add or change the **Group** identifier by typing a new group name in the fields provided.
7. Click **Apply** at the bottom of the page to accept the changes. If you do not click **Apply**, you will lose your changes. If you make a mistake or want to undo any changes, click **Revert**.

Enterprise Manager displays a confirmation page, which confirms that the appropriate configuration files have been updated.

8. Click **Yes** to restart the HTTP Server so the changes will take effect. Click **No** to restart the server later.

Specifying a Port for a Listener

When you start Oracle HTTP Server, it connects to a port and awaits client requests. Oracle HTTP Server will automatically attempt to listen on port 7777.

To specify a listener port:

1. Navigate to the Oracle HTTP Server Home Page on the Oracle Enterprise Manager Web site. Scroll to the Administration section.
2. Select **Server Properties** in the Administration section. This opens the Server Properties Page.

3. Scroll down to the Listening Addresses/Ports table.

The first row in the Listening Addresses/Ports table identifies the default listener port. To edit the default listener port, edit the number in the **Listening Port** column.

To add port settings, click **Add Another Row** to add a new row to the table. Enter the IP address and/or port number to the new row.

4. Scroll down to the end of the page and click **Apply** to accept the changes. If you do not click **Apply**, you will lose your changes. If you make a mistake or want to undo any changes, click **Revert**.

Enterprise Manager displays a confirmation page, which confirms that the appropriate configuration files have been updated.

5. Click **Yes** to restart the HTTP Server so the changes will take effect. Click **No** to restart the server later.

Changing the Error Log Properties

You can change Error Log properties from the Oracle HTTP Server Home Page on the Oracle Enterprise Manager Web site. The Error Log file is an important source of information for maintaining a well-performing server. The Error Log records all of the information about problem situations so that you can easily diagnose and fix the problems. The following directives allow you to customize how your error log is set up.

To customize the error log properties:

1. Navigate to the Oracle HTTP Server Home Page on the Oracle Enterprise Manager Web site. Scroll to the Administration section.
2. Select **Server Properties** in the Administration section. This opens the Server Properties Page.
3. Scroll to the Logging section of the Server Properties Page.
4. Type the full path name of the directory where you want to keep the error log file in the **Error Log Filename** field. You can also type the relative path name. A relative path is assumed to be relative to the Server Root directory.
5. Select the logging level from the **Error Logging Level** drop-down menu. The logging level indicates the severity of the error being reported.

Note: Setting the log level to notice, info, or debug tends to flood the error log with informational messages. Use these options only if you need to perform a very detailed analysis or to debug a specific performance problem.

6. Set the **IP Address Translation** type. This setting tells the server how to handle DNS lookups. Typically a single DNS lookup yields better performance than a double DNS lookup.
7. Scroll to the end of the page and click **Apply** to accept the changes. If you do not click **Apply**, you will lose your changes. If you make a mistake or want to undo any changes, click **Revert**.
Enterprise Manager displays a confirmation page, which confirms that the appropriate configuration files have been updated.
8. Click **Yes** to restart the HTTP Server so the changes will take effect. Click **No** to restart the server later.

Adding an Access Log File

You can change Access Log properties from the Oracle HTTP Server Home Page on the Oracle Enterprise Manager Web site. The Access Log contains basic information about every HTTP transaction that the server handles. Specifically, the access log file contains: hostname, remote logname, remote user, time, request, response code, and bytes transferred. This information can be used to generate statistical reports about the server's usage patterns.

Note: Access Log files are not automatically generated. You must first tell the server where and how to store access log information.

To create an access log file:

1. Navigate to the Oracle HTTP Server Home Page on the Oracle Enterprise Manager Web site. Scroll to the Administration section.
2. Select **Server Properties** in the Administration section. This opens the Server Properties Page.
3. Scroll to the Logging section of the Server Properties Page.

4. Click **Add Another Row** in the Select Access Log table to add a new row. When the page reloads, scroll back to the Logging section.
5. Type the full path and filename of the access log file you want to create in the empty field. For example, you can type the following location:

(UNIX) /private2/ias/Apache/Apache/logs/access_log
(Windows) C:\ias_home\Apache\Apache\logs\access_log

You can also type the relative path and filename. A relative path is assumed to be relative to the Server Root directory:

(UNIX) logs/access_log
(Windows) logs\access_log

6. Set the log format by typing a new format name. The default is common. For information on creating custom log formats, go to "Adding an Access Log File" from the online help for the Server Properties Page.

Note: For a full description of the available log formats, click **Help** at the top of the Server Properties page.

7. Scroll down to the end of the page and click **Apply** to accept the changes. If you do not click **Apply**, you will lose your changes. If you make a mistake or want to undo any changes, click **Revert**.
Enterprise Manager displays a confirmation page, which confirms that the appropriate configuration files have been updated.
8. Click **Yes** to restart the HTTP Server so the changes will take effect. Click **No** to restart the server later.

Changing the Access Log Properties

To change access log properties:

1. Navigate to the Oracle HTTP Server Home Page on the Oracle Enterprise Manager Web site. Scroll to the Administration section.
2. Select **Server Properties** in the Administration section. This opens the Server Properties Page.
3. Scroll to the Logging section of the Server Properties Page.
4. Select the Client Access Log file you want to relocate in the Select Access Log section.
5. Type the new destination in the **Client Access Log Filename** field. The destination can be the full path and filename or a relative path and filename. A relative path is assumed to be relative to the Server Root directory.
6. Scroll to the end of the page and click **Apply** to accept the changes. If you do not click **Apply**, you will lose your changes. If you make a mistake or want to undo any changes, click **Revert**.

Enterprise Manager displays a confirmation page, which confirms that the appropriate configuration files have been updated.

7. Click **Yes** to restart the HTTP Server so the changes will take effect. Click **No** to restart the server later.

Managing the Client Request and Connection Handling

You can specify how the Child Processes and Connections should initialize resources during the server's processing phase through the Oracle HTTP Server Home Page on the Oracle Enterprise Manager Web site. The Child Process and Connection settings impact the ability of the server to process requests. You may need to modify these settings as the number of requests increases or decreases to maintain a well-performing server.

Note: For more information on setting the client request and connection handling parameters, see the *Oracle HTTP Server Administration Guide*.

To modify child process and connection settings:

1. Navigate to the Oracle HTTP Server Home Page on the Oracle Enterprise Manager Web site. Scroll to the Administration section.
2. Select **Server Properties** in the Administration section. This opens the Server Properties Page.
3. Scroll to the Client Request Handling or Client Connection Handling sections of the Server Properties Page.
4. Modify the Client Request Handling and Client Connections Handling directives by changing the default values in the appropriate fields.
For help on individual settings, click Help at the top of the Server Properties page.
5. Scroll down to the end of the page and click **Apply** to accept the changes. If you do not click **Apply**, you will lose your changes. If you make a mistake or want to undo any changes, click **Revert**.
Enterprise Manager displays a confirmation page, which confirms that the appropriate configuration files have been updated.
6. Click **Yes** to restart the HTTP Server so the changes will take effect. Click **No** to restart the server later.

Editing the Server Configuration Files

You can access the HTTP server configuration files directly through the Oracle HTTP Server Advanced Properties Page on the Oracle Enterprise Manager Web site. Use these files to customize the features of your server.

See Also: For more information on using the HTTP server configuration files to customize your server settings, see the *Oracle HTTP Server Administration Guide*.

To edit a configuration file:

1. Navigate to the Oracle HTTP Server Home Page on the Oracle Enterprise Manager Web site. Scroll to the Administration section.
2. Select **Advanced Server Properties** in the Administration section. This opens the Advanced Server Properties Page.
3. Select the configuration file you want to edit. A text editor appears where you can make the appropriate changes.

4. Scroll down to the end of the page and click **Apply** to accept the changes. If you do not click **Apply**, you will lose your changes. If you make a mistake or want to undo any changes, click **Revert**.
Enterprise Manager displays a confirmation page, which confirms that the appropriate configuration files have been updated.
5. Click **Yes** to restart the HTTP Server so the changes will take effect. Click **No** to restart the server later.

Creating a Database Access Descriptor (DAD) for mod_plsql

Oracle HTTP Server contains the `mod_plsql` module, which provide support for building PL/SQL-based applications on the Web. PL/SQL stored procedures retrieve data from a database and generate HTTP responses containing data and code to display in a Web browser.

In order to use `mod_plsql` you must install the PL/SQL Web Toolkit into a database and create a Database Access Descriptor (DAD) which provides `mod_plsql` with connection information for the database.

See Also: "mod_plsql" on page 4-6 for more information on installing the PL/SQL Web Toolkit into a database

You can create a DAD using the Enterprise Manager Web site:

1. Navigate to the HTTP Server Home Page. Scroll to the Administration section.
2. Select **PL/SQL Properties**. This opens the `mod_plsql` Services Page.
3. On the `mod_plsql` Services Page, scroll to the DAD Status section. Click **Create**. This opens the DAD Type Page.
4. If you intend to use `mod_plsql` with Oracle9iAS Portal or Oracle Login Server, select the **Portal** radio button. Otherwise, select the **General** radio button. The subsequent screens will be populated with default values based on your selection. Click **Next**. This opens the Database Connection Page.
5. Type a unique name in the **DAD Name** field. Enter the database account, password, and connection information in the Database Connectivity Information section. In the **Default Page** field, type the name of the PL/SQL procedure that should be invoked when one is not specified. In the **NLS Language** field, type the Oracle Language and Character Set for the back-end database. Choose an **Authentication Mode** in the Authentication Mode section. Click **Next**. This opens the Document, Alias, and Session Page.

6. On the Document, Alias, and Session Page, fill in the fields that are required for your DAD configuration. Refer to the online help for more information. Click **Next**. This opens the Advanced Page.
7. On the Advanced Page, fill in the fields that are required for your DAD configuration. These fields are typically not configured. Refer to the online help for more information. Click **Finish**. This opens the Confirmation Page. Click **OK**.
8. Restart Oracle HTTP Server.

See Also: For more details on mod_plsql configuration parameters, refer to "mod_plsql" in Chapter 6 of *Oracle HTTP Server Administration Guide*

Deleting a Database Access Descriptor (DAD) for mod_plsql

You can delete a DAD using the Enterprise Manager Web site:

1. Navigate to the HTTP Server Home Page. Scroll to the Administration section.
2. Select **PL/SQL Properties**. This opens the mod_plsql Services Page.
3. On the mod_plsql Services Page, scroll to the DAD Status section. Select the radio button in the Select column for the DAD you would like to delete. Click **Delete**.
4. Restart Oracle HTTP Server.

Enabling JServ

It is recommended that you use the Oracle9iAS Containers for J2EE (OC4J) for your servlet environment; it is the default configuration for Oracle9iAS Release 2 (9.0.2). However, you may want or need to use JServ in your Oracle9iAS Release 2 installation. These instructions are provided to explain how to enable JServ, and, if necessary, use it for some applications and OC4J for others. The instructions assume a working familiarity with the Oracle HTTP Server and JServ configuration.

This section has the following topics:

- [Directives Reference](#)

This section describes directives used to enable JServ with mod_oprocmgr.

- [Enabling JServ with mod_oprocmgr](#)

This section explains how to enable the Oracle default mode for JServ. Use this mode if you want process management and load balancing capabilities for multiple JVMs. The `ApJServManual` directive has a new mode, 'auto', that enables using JServ with the Oracle module `mod_oprocmgr`. The following file contains `LoadModule` directives for `mod_jserv` and `mod_oprocmgr`:

(UNIX) `ORACLE_HOME/Apache/Jserv/etc/jserv.conf`
(Windows) `ORACLE_HOME\Apache\Jserv\conf\jserv.conf`

- [Enabling JServ in Automatic Mode](#)

This section explains how to enable JServ in automatic mode. Use this mode if you need only one JVM. In this mode, the `ApJServManual` directive is set to 'off' and the `mod_jserv` module launches and monitors the JVM. If the Oracle HTTP Server is restarted or stopped, `mod_jserv` restarts or stops the JVM.

- [Enabling JServ in Manual Mode](#)

This section explains how to enable JServ in manual mode. Use this mode if you need to run multiple JVMs. In this mode, the `ApJServManual` directive is set to 'on' and you have to stop and start the JVM manually. To monitor the JVM, you must use an external monitoring facility.

- [Using JServ and OC4J Together](#)

This section explains how to use `mod_rewrite` to enable some applications to execute on JServ, and others on OC4J.

Directives Reference

This section describes some JServ configuration directives that are related to using JServ with `mod_oprocmgr`. All directives are described thoroughly in the configuration files.

See Also: *Oracle HTTP Server Administration Guide*

Directives in `jserv.properties`

This section describes directives in `jserv.properties`.

Port

Use this directive to specify the ports to which JServ will bind. For example:

```
port=8007
```

If no ports are specified, the JServ processes will choose their ports. The following port directive enables the JServ processes to choose their own ports (the default behavior):

```
port=
```

If you eliminate the port directive entirely, an error will occur.

You can specify multiple ports, and separate the values with commas. Note that a range of ports (9000-9010) is a valid value.

```
port=8007,9000-9010,8010
```

Directives in jserv.conf

This section describes directives in `jserv.conf`.

ApJServManual This directive accepts a new mode, `auto`, which invokes the new infrastructure functionality (in which `mod_oprocmgr` manages processes). The syntax is:

```
ApJServManual auto
```

You can set the mode to `on` or `off` to use standard JServ functionality.

ApJServGroup This directive defines groups for the process manager to manage for `mod_jserv`. If you have worked with `mod_jserv`, you will note that this directive replaces the `ApJServBalance`, `ApJServHost`, `ApJServRoute` and `ApJServShmFile` directives.

All JServ processes to be managed must belong to a group, and each group has its own `ApJServGroup` directive. If you only have one JServ process, you must define a group with just that process in it. The processes in a group are identical except for their listening ports, so requests directed to the group are distributed evenly among the processes.

The `ApJServGroup` directive takes four arguments: `groupname`, `number of processes`, `node weight`, and `properties file`. In the following example, the `groupname` is `mygroup`, the `number of processes` is `1`, the `node weight` is `1`, and the full path of the `properties` file used to start the JServ processes is:

```
(UNIX) /private/ias/Apache/Jserv/etc/jserv.properties  
(Windows) C:\iashome\Apache\Jserv\conf\jserv.properties
```

```
(UNIX) ApJServGroup mygroup 1 1 /private/ias/Apache/Jserv/etc/jserv.properties  
(Windows) ApJServGroup mygroup 1 1 C:\iashome\Apache\Jserv\conf\jserv.properties
```

ApJServGroupMount This directive defines a mount point and maps it to a process group and zone. In the following example, the mount point is /servlets, the group is mygroup, and the zone is root. Note that the balance protocol is in use for routing, as in the standard JServ configuration.

```
(UNIX) ApJServGroupMount /servlets balance://mygroup/root  
(Windows) ApJServGroupMount /servlets balance://mygroup/root
```

Place this directive after the **ApJServGroup** directive in the configuration file.

ApJServGroupSecretKey This directive specifies the secret key that JServ needs to authenticate clients. It can be disabled, as follows:

```
ApJServGroupSecretKey disabled
```

When activated, the directive takes one or two arguments. In the following example, with group and filename arguments, the filename `mysecretkey` applies to the group `mygroup`:

```
(UNIX) ApJServGroupSecretKey mygroup /usr/local/apache/jserv/mysecretkey  
(Windows) ApJServGroupSecretKey mygroup C:\ias\apache\jserv\mysecretkey
```

You can supply only the filename argument. No group is named, so the secret key filename applies to all groups.

```
(UNIX) ApJServGroupSecretKey /usr/local/apache/jserv/mysecretkey  
(Windows) ApJServGroupSecretKey C:\ias\apache\jserv\mysecretkey
```

You cannot combine directives using the one-argument syntax with directives using the two-argument syntax. If you use the two-argument syntax, the default for groups without a group-specific secret key is 'disabled'.

Place this directive after the **ApJServGroup** directive in the configuration file.

Warning: The secret in the secret key file specified in **ApJServSecretKey** must be the same as that specified by the `security.secretKey` directive in the `jserv.properties` file. If the secrets are not the same, the death detection mechanism assumes that all the servlet engine processes are dead, eliminates them, and starts new processes to replace them (repeating the cycle endlessly).

Enabling JServ with mod_oprocmgr

This section explains how to implement process management and load balancing services for JServ processes with mod_oprocmgr. Terms used in this section to describe the module and its functions are defined in [Table 6-2](#):

Table 6-2 mod_oprocmgr Terms and Definitions

Term	Definition
mod_oprocmgr	A module that starts, stops, and detects death of processes (starting new processes to replace them), and provides load balancing services to the processes. mod_oprocmgr gets the topology management information via HTTP requests from internal servers such as JServ, and does its job based on this information.
group	A set of processes across which request traffic is distributed.
servlet engine process	A JVM instance that runs a servlet engine, such as JServ.

How mod_oprocmgr Works with mod_jserv

mod_oprocmgr provides infrastructure capabilities, such as automatic starting of processes, death detection and restart, and load balancing. These capabilities are enabled by a new mode, auto, for the ApJServManual directive.

Based on the configuration information provided by mod_jserv, mod_oprocmgr starts the specified number of JServ processes, managing them for the life of the servers.

Follow these steps to enable JServ with mod_oprocmgr:

1. Uncomment the Include directive for the jserv.conf file in

```
(UNIX) ORACLE_HOME/Apache/Apache/conf/httpd.conf  
(Windows) ORACLE_HOME\Apache\Apache\conf\httpd.conf
```

by removing the initial '#' character in the line:

```
(UNIX) include "/ORACLE_HOME/Apache/Jserv/etc/jserv.conf"  
(Windows) include "C:\ORACLE_HOME\Apache\Jserv\conf\jserv.conf"
```

2. Configure directives in the following file, if needed:

(UNIX) *ORACLE_HOME*/Apache/Jserv/etc/jserv.conf
(Windows) *ORACLE_HOME*\Apache\Jserv\conf\jserv.conf

3. Configure directives in the following file, if needed:

(UNIX) *ORACLE_HOME*/Apache/Jserv/etc/jserv.properties
(Windows) *ORACLE_HOME*\Apache\Jserv\conf\jserv.properties

4. Configure directives in the following file, if needed:

(UNIX) *ORACLE_HOME*/Apache/Jserv/etc/zone.properties
(Windows) *ORACLE_HOME*\Apache\Jserv\conf\zone.properties

5. Configure JServ using the Enterprise Manager Web site:

- a. Navigate to the Instance Home Page on the Enterprise Manager Web site. Scroll to the Administration section.
- b. Select **Configure Components**. This opens the Configure Components Page.
- c. Choose JServ in the Component drop-down menu, enter the `ias_admin` password, and click **OK**.

6. Restart Oracle HTTP Server.

Enabling JServ in Automatic Mode

Follow these steps to enable JServ in automatic mode:

1. Uncomment the Include directive for the jserv.conf file in

(UNIX) *ORACLE_HOME*/Apache/Apache/conf/httpd.conf
(Windows) *ORACLE_HOME*\Apache\Apache\conf\httpd.conf

by removing the initial '#' character in the line:

(UNIX) `include "/ORACLE_HOME/Apache/Jserv/etc/jserv.conf"`
(Windows) `include "C:\ORACLE_HOME\Apache\Jserv\conf\jserv.conf"`

2. Set the ApJServManual directive to off

`ApJServManual off`

in following file:

(UNIX) *ORACLE_HOME*/Apache/Jserv/etc/jserv.conf
(Windows) *ORACLE_HOME*\Apache\Jserv\conf\jserv.conf

3. Configure other directives as needed in `jserv.conf`.
4. Set the port directive in the following file:

(UNIX) `ORACLE_HOME/Apache/Jserv/etc/jserv.properties`
(Windows) `ORACLE_HOME\Apache\Jserv\conf\jserv.properties`

to the same value as that specified in the `ApJServDefaultPort` directive.

5. Configure directives in the following file, if needed:

(UNIX) `ORACLE_HOME/Apache/Jserv/etc/zone.properties`
(Windows) `ORACLE_HOME\Apache\Jserv\conf\zone.properties`

6. Configure JServ using the Enterprise Manager Web site:
 - a. Navigate to the Instance Home Page on the Enterprise Manager Web site. Scroll to the Administration section.
 - b. Select **Configure Components**. This opens the Configure Components Page.
 - c. Choose JServ in the Component drop-down menu, enter the `ias_admin` password, and click **OK**.
7. Restart Oracle HTTP Server.

Enabling JServ in Manual Mode

Follow these steps to enable JServ in manual mode:

1. Uncomment the Include directive for the `jserv.conf` file in

(UNIX) `ORACLE_HOME/Apache/Apache/conf/httpd.conf`
(Windows) `ORACLE_HOME\Apache\Apache\conf\httpd.conf`

by removing the initial '#' character in the line:

(UNIX) `include "/ORACLE_HOME/Apache/Jserv/etc/jserv.conf"`
(Windows) `include "C:\ORACLE_HOME\Apache\Jserv\conf\jserv.conf"`

2. Set the `ApJServManual` directive to on:

`ApJServManual on`

in following file:

(UNIX) `ORACLE_HOME/Apache/Jserv/etc/jserv.conf`

(Windows) *ORACLE_HOME\Apache\Jserv\conf\jserv.conf*

3. Configure other directives as needed in *jserv.conf*.
4. Configure directives in the following file, if needed:

(UNIX) *ORACLE_HOME/Apache/Jserv/etc/jserv.properties*

(Windows) *ORACLE_HOME\Apache\Jserv\conf\jserv.properties*

5. Configure directives in the following file, if needed:
(UNIX) *ORACLE_HOME/Apache/Jserv/etc/zone.properties*
(Windows) *ORACLE_HOME\Apache\Jserv\conf\zone.properties*

6. Start JServ:

(UNIX) *ORACLE_HOME/Apache/Apache/bin/startJServ.sh*

(Windows) *ORACLE_HOME\Apache\Apache\bin\startJsv.bat*

7. Configure JServ using the Enterprise Manager Web site.
 - a. Navigate to the Instance Home Page on the Enterprise Manager Web site. Scroll to the Administration section.
 - b. Select **Configure Components**. This opens the Configure Components Page.
 - c. Choose JServ in the Component drop-down menu, enter the *ias_admin* password, and click **OK**.
8. Restart Oracle HTTP Server.

Using JServ and OC4J Together

Perform the following configuration steps to enable JServ and Oracle9iAS Containers for J2EE (OC4J) to coexist. This is important if you have the Portal and Wireless installation type, because of the Portal dependency on OC4J.

1. Specify the engine on which applications should execute. Suppose you have these URLs:

- */application1/file1.jsp* to execute on JServ
- */application2/file2.jsp* to execute on OC4J

You must rewrite the URL for application1.

- a. Edit the following file:

(UNIX) *ORACLE_HOME*/Apache/Apache/conf/httpd.conf
(Windows) *ORACLE_HOME*\Apache\Apache\conf\httpd.conf

and ensure that the following directives are active (uncommented) and present:

```
LoadModule rewrite_module libexec/mod_rewrite.so
AddModule mod_rewrite.c
RewriteEngine on
```

b. Edit the following file:

(UNIX) *ORACLE_HOME*/Apache/jsp/conf/ojsp.conf
(Windows) *ORACLE_HOME*\Apache\jsp\conf\ojsp.conf

to add these directives:

```
RewriteRule /application1/(.*)/(.*).jsp$ /application1/$1/$2.jsp1
ApJServAction .jsp1 /servlets/oracle.jsp.JspServlet
```

c. Remove this directive:

```
ApJServAction .jsp /servlets/oracle.jsp.JspServlet
```

d. Edit the following file:

(UNIX) *ORACLE_HOME*/Apache/Jserv/etc/jserv.conf
(Windows) *ORACLE_HOME*\Apache\Jserv\conf\jserv.conf

and mount /servlets to the JVM that will service the JSP requests. Use the ApJServMount or ApJServGroupMount directive (depending on how the JServ processes are started).

2. Configure JServ using the Enterprise Manager Web site:
 - a. Navigate to the Instance Home Page on the Enterprise Manager Web site. Scroll to the Administration section.
 - b. Select **Configure Components**. This opens the Configure Components Page.
 - c. Choose JServ in the Component drop-down menu, enter the `ias_admin` password, and click **OK**.
3. Restart Oracle HTTP Server.

Configuring Oracle9iAS Containers for J2EE

You can configure Oracle9iAS Containers for J2EE (OC4J) using the Enterprise Manager Web site. You can use the Instance Home Page to create and delete OC4J instances, each of which has its own OC4J Home Page. You can use each individual OC4J Home Page to configure the corresponding OC4J instance and its deployed applications.

This section provides instructions for creating and deleting OC4J instances and an overview of the configuration tasks you can perform on OC4J instances and applications.

See Also: *Oracle9iAS Containers for J2EE User's Guide* for more detailed instructions on configuring OC4J instances and applications

Creating an OC4J Instance

Every application server instance has a default OC4J instance named `OC4J_home`. You can create additional instances, each with a unique name, within an application server instance.

To create a new OC4J instance:

1. Navigate to the Instance Home Page on the Oracle Enterprise Manager Web site. Scroll to the System Components section.
2. Click **Create OC4J Instance**. This opens the Create OC4J Instance Page.
3. In the Create OC4J Instance Page, type a unique instance name in the OC4J instance name field. Click **Create**.

A new OC4J instance is created with the name you provided. This OC4J instance shows up on the Instance Home Page in the System Components section. The instance is initially in a stopped state and can be started any time after creation.

Deleting an OC4J Instance

To delete an OC4J instance:

1. Navigate to the Oracle9iAS Instance Home Page on the Oracle Enterprise Manager Web site. Scroll to the System Components section.
2. Select the radio button in the Select column for the OC4J instance you would like to delete.
3. Click **Delete**.

Overview of OC4J Configuration Tasks

You can use the Enterprise Manager Web site to do the following types of configuration:

- [OC4J Instance Level Configuration](#)
- [Application Level Configuration](#)

OC4J Instance Level Configuration

Each OC4J instance has its own OC4J Home Page which allows you to configure global services and deploy applications to that instance.

Some examples of the tasks you can perform from the OC4J Home Page are:

- Configure Web Site
- Configure Global JSP Container Parameters
- Configure Global Web Application Parameters
- Configure RMI and JMS
- Configure Data Sources
- Configure Security
- Configure UDDI Registry and JAAS Parameters
- Manipulate XML Files

Application Level Configuration

Once you have deployed an application, it has its own Application Home Page, which is accessible from the OC4J Home Page. Using the Application Home Page, you can modify most of the application parameters. Some examples of the tasks you can perform from the Application Home Page are:

- Configure Application General Parameters
- Configure Local J2EE Services
- Modify XML Files Included in the Deployed Application EAR File

See Also: *Oracle9iAS Containers for J2EE User's Guide* contains complete instructions for configuring OC4J instances and applications using the Enterprise Manager Web site

Configuring Oracle9iAS Web Cache

Oracle9iAS Web Cache is an example of a component that has integrated a pre-Oracle9iAS Release 2 configuration tool into the Enterprise Manager Web site. This simplifies administration by providing access to the management features of the Oracle9iAS Web Cache Manager and the management features of the Enterprise Manager Web site from a single Home Page.

To access Oracle9iAS Web Cache Manager:

1. Navigate to the Instance Home Page.
2. Select the Web Cache instance you want to configure in the Name column. This displays the Web Cache Home Page.
3. From the Administration section, select Web Cache Administration.

See Also: For information on using Oracle9iAS Web Cache Manager, refer to *Oracle9iAS Web Cache Administration and Deployment Guide*.

Component Configuration Dependencies

If you change a configuration value for a component, for example, the port number for Web Cache or the host for Oracle Internet Directory, your change may affect other components. You may need to update parameters in other components to ensure that all the components still function.

Each section in this chapter describes a component and its parameters that can affect other components.

J2EE and Web Cache install type:

- [Oracle HTTP Server](#)
- [Web Cache](#)
- [OPMN](#)

Portal and Wireless install type:

- [Portal](#)
- [Web Providers \(for Portals\)](#)
- [Provider Groups \(for Portals\)](#)
- [Wireless](#)

Infrastructure

- [Oracle9iAS Single Sign-On](#)
- [Oracle Internet Directory](#)
- [Metadata Repository](#)

Oracle HTTP Server

Parameters for Oracle HTTP Server are defined in .conf files in the *ORACLE_HOME/Apache* area. For example:

- *ORACLE_HOME/Apache/Apache/conf/httpd.conf*
- Module configuration files, such as:
 - *ORACLE_HOME/Apache/Apache/conf/mod_oc4j.conf*
 - *ORACLE_HOME/Apache/modplsql/conf/dads.conf*
 - *ORACLE_HOME/Apache/modplsql/conf/plsql.conf*
 - *ORACLE_HOME/Apache/oradav/conf/moddav.conf*

Note: You need to use Oracle Enterprise Manager Web site to edit the configuration files. If you edit them manually, you need to run `dcmctl` with the `updateConfig` parameter so that the updated values are copied into the DCM repository. See "[About Manually Editing HTTP Server and OC4J Configuration Files](#)" on page 6-2 for details.

For detailed information on Oracle HTTP Server and its parameters, see the *Oracle HTTP Server Administration Guide*.

Table 7-1 Changes to Oracle HTTP Server

If you change this:	You affect these components:
Listener port number (Port or Listen directive in <code>httpd.conf</code>) or Host (<code>ServerName</code> directive in <code>httpd.conf</code>) or Protocol (HTTP or HTTPS)	Oracle9iAS Web Cache: <ol style="list-style-type: none">1. Update the port number in the Application Web Servers section under the General Configuration section of the Web Cache Home Page.2. Restart Web Cache. Oracle Internet Directory: Change the <code>orcldasurlbase</code> attribute in the "cn=OperationURLs, cn=DAS, cn=Products, cn=OracleContext" entry to reflect the new port number in the URL. The value of the attribute is the Oracle HTTP Server's URL (for example, <code>http://mymachine.oracle.com:7777/</code>). You can make the change using Oracle Internet Directory Manager (see the <i>Oracle Internet Directory Administrator's Guide</i> for details).
	(continued in next row)

Table 7–1 Changes to Oracle HTTP Server (Cont.)

If you change this:	You affect these components:
(continued from previous row)	
Listener port number (Port or Listen directive in httpd.conf)	Oracle9iAS Single Sign-On: If you change the host, port, or protocol for Oracle HTTP Server, you need to update these values for Oracle9iAS Single Sign-On. When you make these changes in Single Sign-On, you affect the partner applications (Portal, Wireless, and mod_osso) in Single Sign-On. You then have to re-register the partner applications with Single Sign-On using the new host, port, or protocol.
or	
Host (ServerName directive in httpd.conf)	
or	
Protocol (HTTP or HTTPS)	<p>1. Update the hostname, port, or protocol data in the Oracle9iAS Single Sign-On schema on the Infrastructure machine.</p> <p>You do this by running the script:</p> <p>(UNIX) <code>ORACLE_HOME/sso/bin/ssocfg.sh</code></p> <p>(Windows) <code>ORACLE_HOME\sso\bin\ssocfg.bat</code></p> <p>on the Infrastructure machine to update tables in the Oracle9iAS Single Sign-On schema with the new data. The script has the following syntax:</p> <p><code>ssocfg.{sh,bat} protocol host port</code></p> <p>Before running the command:</p> <p>(UNIX and Windows) You need to set <code>ORACLE_HOME</code> and <code>ORACLE_SID</code>.</p> <p>(UNIX) You need to include <code>\$ORACLE_HOME/lib</code> in the <code>LD_LIBRARY_PATH</code> environment variable.</p> <p>(Windows) You need to include <code>%ORACLE_HOME%\lib</code> and <code>%ORACLE_HOME%\bin</code> in the <code>PATH</code> environment variable.</p> <p>For details on the script, see Chapter 2, "Administering Oracle Single Sign-On," of the <i>Oracle9iAS Single Sign-On Administrator's Guide</i>. Look in the "Enabling the Single Sign-On Server for SSL" section.</p> <p>2. Re-register Oracle9iAS Single Sign-On partner applications (Portal, Wireless, mod_osso) by running their registration tools. The tool varies for each partner application:</p> <p>For Portal, the registration tool is <code>ptlasst.csh</code> for UNIX, <code>ptlasst.cmd</code> for Windows.</p> <p>For Wireless, the registration tool is <code>reRegistersSO.sh</code> for UNIX, <code>reRegistersSO.bat</code> for Windows.</p> <p>For mod_osso, the registration tool is packaged in <code>ossoreg.jar</code>.</p> <p>See the section "Oracle9iAS Single Sign-On" for details.</p>

(continued in next row)

Table 7–1 Changes to Oracle HTTP Server (Cont.)

If you change this:	You affect these components:
(continued from previous row)	Oracle9iAS Single Sign-On (continued):
Listener port number (Port or Listen directive in httpd.conf)	<p>3. Remove the old partner application using the Administer Partner Applications page. Click the "x" in the Delete column to delete the application.</p>
or	<p>You need to delete the application because when you re-register a partner application, the registration tool creates a new entry for the partner application instead of updating the existing application.</p>
Host (ServerName directive in httpd.conf)	<p>You can determine which application is the older one by looking at the application ID (click Edit to view details for the application -- applications with low IDs were created first). You can also look at the URLs used by the application. If an application uses an incorrect host, port, or protocol for its URLs, you should delete the application.</p>
or	<p>4. Modify the <code>HTTPMachine</code> property for the Oracle9iAS Single Sign-On target in the file <code>ORACLE_HOME/sysman/emd/targets.xml</code> to match the values used by Oracle HTTP Server.</p>
Protocol (HTTP or HTTPS)	<p>You do not need to modify the <code>HTTPPort</code> and <code>HTTPProtocol</code> properties. If the port changes, the code detects the new port number and updates the <code>HTTPPort</code> property for the Oracle9iAS Single Sign-On target in the <code>targets.xml</code> file.</p>
	<p>If the name of the machine running the metadata repository database changes, you need to update the <code>ssoServerMachineName</code> property for the Oracle9iAS Single Sign-On target in the file <code>ORACLE_HOME/sysman/emd/targets.xml</code>.</p>
	<p>Note that if you have multiple repository database instances, Oracle9iAS Single Sign-On uses only one of the instances. You need to update the <code>ssoServerMachineName</code> property to point to the machine that hosts that instance.</p>
	<p>Oracle9iAS Wireless:</p>
	<ol style="list-style-type: none">1. In the Wireless site configuration section of Oracle Enterprise Manager Web site, update the URL to reflect the new host and port settings.2. Re-register the Wireless partner application. See the section ""Oracle9iAS Single Sign-On" for details.
	<i>(continued in next row)</i>

Table 7–1 Changes to Oracle HTTP Server (Cont.)

If you change this:	You affect these components:
(continued from previous row)	Oracle9iAS Portal:
Listener port number (Port or Listen directive in httpd.conf)	1. Modify the HTTPPort property for the Portal target in the file <i>ORACLE_HOME/sysman/ems/targets.xml</i> . 2. Modify the HTTPPort property for the Oracle HTTP Server target in the file <i>ORACLE_HOME/sysman/ems/targets.xml</i> .
or	
Host (ServerName directive in httpd.conf)	3. To create a new association of Portal with the Oracle9iAS Single Sign-On server, with the modified port, run the script: (UNIX) <i>ORACLE_HOME/assistants/opca/ptlasst.csh</i> (Windows) <i>ORACLE_HOME\assistants\opca\ptlasst.cmd</i> If Web Cache is used, run the script in midtier mode (that is, <i>-mode MIDTIER</i>). If Web Cache is not used, run the script in <i>SSOPARTNERCONFIG</i> mode (that is, <i>-mode SSOPARTNERCONFIG</i>). For details on <i>ptlasst</i> , see the <i>Oracle9iAS Portal Configuration Guide</i> .
Protocol (HTTP or HTTPS)	
	mod_osso: Re-register the mod_osso partner application with Single Sign-On. See the section " Oracle9iAS Single Sign-On " for details.
	Oracle Enterprise Manager: Create a virtual host with the original hostname. You can do this by adding the following lines to httpd.conf :
	<pre>NameVirtualHost * <VirtualHost *> ServerName original_hostname DocumentRoot /path_to_default_document_root </VirtualHost></pre>
	Oracle9iAS Reports Services: Update your reports links.
	Oracle9iAS Discoverer: If the hostname or port for the HTTP Server is changed, then it affects the Oracle9iAS Discoverer Portlet Provider. You should update the registration of the Discoverer Portlet Provider with Oracle Portal. For details, see the <i>Oracle9iAS Discoverer Configuration Guide</i> .
	(continued in next row)

Table 7–1 Changes to Oracle HTTP Server (Cont.)

If you change this:	You affect these components:
(continued from previous row)	OPMN:
Listener port number (Port or Listen directive in httpd.conf)	No changes required.
or	
Host (ServerName directive in httpd.conf)	JServ: No changes required.
or	
Protocol (HTTP or HTTPS)	
Log directory (CustomLog directive in httpd.conf)	Oracle9iAS Clickstream Intelligence: Update the logFileLoc property in agent.xml.
Port for OC4J	Oracle9iAS Containers for J2EE: The host and port used for the Oc4jMount directive must match the AJP host and port for the OC4J being routed to. The OC4J configuration is specified in the file xxx-web-site.xml (for example, http-web-site.xml).
Note: Changing this directive is not recommended.	
Any	mod_oradav: If you make any configuration changes in Oracle HTTP Server, you should not have to make any changes in moddav.conf because mod_oradav uses environment variables that are also used for OCI clients (LD_LIBRARY_PATH, ORACLE_HOME, NLS_LANG, TNS_ADMIN, etc).

Web Cache

You manage Web Cache using Oracle Web Cache Manager, which is a Web-based administration tool. You can use this tool to change the various Web Cache port numbers and passwords. See the *Oracle9iAS Web Cache Administration and Deployment Guide* for a full description of the tool.

Table 7–2 Changes to Web Cache

If you change this:	You affect these components:
Administration port	Oracle9iAS Portal: Run the script: (UNIX) <code>ORACLE_HOME/assistants/opca/ptlasst.csh</code> (Windows) <code>ORACLE_HOME\assistants\opca\ptlasst.cmd</code> in midtier mode (that is, <code>-mode MIDTIER</code>) and specify the new administration port value in the <code>-cpport_a</code> parameter. For details on <code>ptlasst</code> , see the <i>Oracle9iAS Portal Configuration Guide</i> .

Table 7–2 Changes to Web Cache (Cont.)

If you change this:	You affect these components:
Listener port or machine running Web Cache	<p>Oracle9iAS Wireless: In the Wireless site configuration section of Oracle Enterprise Manager Web site, update the URL to reflect the new host and port settings.</p> <p>Oracle9iAS Reports Services: Manually update your reports links.</p> <p>Oracle9iAS Portal:</p> <ol style="list-style-type: none"> Run the script: (UNIX) <code>ORACLE_HOME/assistants/opca/ptlasst.csh</code> (Windows) <code>ORACLE_HOME\assistants\opca\ptlasst.cmd</code> in midtier mode (that is, <code>-mode MIDTIER</code>) and specify the new port and hostname values using the appropriate parameters: <ul style="list-style-type: none"> - new listener port in the <code>-port</code> parameter - new listener hostname in the <code>-host</code> parameter - new cache hostname in the <code>-chost</code> parameter For details on <code>ptlasst</code>, see the <i>Oracle9iAS Portal Configuration Guide</i>. Update the <code>Port</code> or <code>ServerName</code> directive in the file <code>httpd.conf</code> for Oracle HTTP Server, as appropriate. Restart Oracle HTTP Server. <p>Portal (JPDK): Modify the configuration file for the JPDK invalidation API.</p>
Invalidation port	<p>Oracle9iAS Wireless: In the Wireless site configuration section of Oracle Enterprise Manager Web site, update the Web Cache Invalidation Port field to reflect the new value.</p> <p>Oracle9iAS Portal: Run the script: (UNIX) <code>ORACLE_HOME/assistants/opca/ptlasst.csh</code> (Windows) <code>ORACLE_HOME\assistants\opca\ptlasst.cmd</code> in midtier mode (that is, <code>-mode MIDTIER</code>) and specify the new invalidation port value in the <code>-cport_i</code> parameter. For details on <code>ptlasst</code>, see the <i>Oracle9iAS Portal Configuration Guide</i>.</p> <p>Portal (JPDK): Modify the configuration file for the JPDK invalidation API.</p>

Table 7–2 Changes to Web Cache (Cont.)

If you change this:	You affect these components:
Invalidation password	<p>Oracle9iAS Wireless:</p> <p>In the Wireless site configuration section of Oracle Enterprise Manager Web site, update the Web Cache Invalidation Password field to reflect the new value.</p> <p>Oracle9iAS Portal:</p> <p>Run the script:</p> <p>(UNIX) <code>ORACLE_HOME/assistants/opca/ptlasst.csh</code></p> <p>(Windows) <code>ORACLE_HOME\assistants\opca\ptlasst.cmd</code></p> <p>in midtier mode (that is, <code>-mode MIDTIER</code>) and specify the new invalidation password in the <code>-wc_i_pwd</code> parameter.</p> <p>For details on <code>ptlasst</code>, see the <i>Oracle9iAS Portal Configuration Guide</i>.</p>
Web Cache timeout (OSRECV_TIMEOUT parameter in webcache.xml)	<p>Oracle9iAS Wireless:</p> <p>In the Wireless site configuration section of Oracle Enterprise Manager Web site, update the Web Cache Timeout field to reflect the new value.</p> <p>Oracle9iAS Portal:</p> <p>Update the PPE (Portal Parallel Page Engine) <code>minTimeout</code> parameter in the file <code>ORACLE_HOME/j2ee/OC4J_Portal/applications/portal/portal/WEB-INF/web.xml</code>. Note that the Web Cache timeout must be longer than the PPE timeout.</p>

OPMN

Oracle Process Management Notification (OPMN) keeps track of Oracle9*i*AS processes. It uses the configuration file *ORACLE_HOME*/opmn/conf/opmn.xml.

Table 7–3 Changes to OPMN

If you change this:	You affect these components:
instanceName attribute of the oc4j element in the opmn.xml	Oracle HTTP Server (mod_oc4j): You need to update the OC4J instance name in the Oc4jMount directive in mod_oc4j.conf.

Portal

The Portal feature in Oracle9iAS enables you to aggregate pages from different data sources and display them on a single page. Users can customize the pages that they want to see on their page. For details on Portals, see the *Oracle9iAS Portal Configuration Guide*.

Table 7–4 Changes to Portal

If you change this:	You affect these components:
Mobile gateway URL	<p>Oracle9iAS Wireless:</p> <p>Run the script:</p> <p>(UNIX) <code>ORACLE_HOME/wireless/sample/portalRegistrar.sh</code></p> <p>(Windows) <code>ORACLE_HOME\wireless\sample\portalRegistrar.bat</code></p> <p>to reregister the mobile gateway parameter with Portal.</p> <p>Syntax:</p> <pre>portalregistrar.{sh,bat} <wireless_admin_user> <new_wireless_portal_URL></pre> <p>Example:</p> <pre>portalregistrar.sh orcladmin http://new.myCompany.com/ptg/rm</pre>
Portal database schema password (using Portal UI or SQL*Plus)	<p>HTTP Server (mod_plsql):</p> <ol style="list-style-type: none"> 1. Use Oracle Enterprise Manager Web site to change the Database Password field for the DAD that accesses the Portal schema to match the new password. <p>This changes the <code>PlsqlDatabasePassword</code> parameter in the <code>ORACLE_HOME/Apache/modplsql/conf/dad.conf</code> file.</p> <ol style="list-style-type: none"> 2. Restart HTTP Server. <p>Oracle Internet Directory and DIP (Directory Integration Profile) Synchronization:</p> <p>Run the script:</p> <p>(UNIX) <code>ORACLE_HOME/bin/oidprovtool</code></p> <p>(Windows) <code>ORACLE_HOME\bin\oidprovtool.bat</code></p> <p>to modify (or delete or recreate) the Portal's provisioning profile to update the password that is maintained in Oracle Internet Directory so the DIP agent can connect to the portal to send the notifications.</p>

Table 7–4 Changes to Portal (Cont.)

If you change this:	You affect these components:
Moving Portal to a different database	Oracle Enterprise Manager: Add the new database target to the <i>ORACLE_HOME/sysman/emd/targets.xml</i> file and update <i>iasDatabaseTargetName</i> in <i>targets.xml</i> in the portal target instance to point to the new database.
Portal port, DAD name, URL of Home Page	Oracle9iAS Wireless and Oracle9iAS Single Sign-On: <ol style="list-style-type: none">1. Run the script: (UNIX) <i>ORACLE_HOME/assistants/opca/ptlasst.csh</i> (Windows) <i>ORACLE_HOME\assistants\opca\ptlasst.cmd</i> to update Oracle9iAS Single Sign-On. This also updates the Portal repository with the host, port, and DAD information. For details on <i>ptlasst</i>, see the <i>Oracle9iAS Portal Configuration Guide</i>.2. Use the Wireless administration tool to identify the service that represents Portal, and update the URL associated with the service.

Web Providers (for Portals)

A Web provider owns portlets that can be placed on portal pages. Web provider details are maintained using the Add/Edit Provider screens in the portal. These pages are used to maintain both Web Providers and Database Providers.

Table 7–5 Changes to Portal Web Service Provider

If you change this:	You affect these components:
Web provider URL or port	Oracle9iAS Portal: Update provider registration information with new URL or port.
Web provider service ID	Oracle9iAS Portal: Update provider registration information with new service ID.
Web provider mount point	Oracle9iAS Portal: Update provider registration information with new mount point.
Adding Web Cache in front of Web provider	Oracle9iAS Portal: Do one of the following: <ul style="list-style-type: none"> ▪ Configure Web Cache so it has the hostname/port of Web provider listener and assign a new host/port to Web provider. OR ▪ Update provider registration information with the new URL, then create a Web Cache configuration file and associate the file with OC4J so that the Web provider can send invalidation requests to Web Cache. The location of the configuration file is a Java system parameter. See the <i>Oracle9iAS Web Cache Administration and Deployment Guide</i> for details.
Removing Web Cache from Web provider	Oracle9iAS Portal: Do one of the following: <ul style="list-style-type: none"> ▪ Configure the Web provider listener to point to the host and port of the origin HTTP Server. OR ▪ Update the provider registration information with the new URL.
Moving Web Cache to a different host or port	Oracle9iAS Portal: Update the Web Cache configuration file with the new host or port.
Changing the service name or location of the database that stores provider preference	Oracle9iAS Containers for J2EE: Update the OC4J <code>data-sources.xml</code> file with the new database location or service name.

Table 7–5 Changes to Portal Web Service Provider (Cont.)

If you change this:	You affect these components:
Changing domain of portal or Web provider	Oracle9iAS Portal: If you have applications that rely on "same cookie domain" (where portal delegates Web provider cookie management to the browser), you have to change the scope of the Web provider cookie so that both portal and Web provider listeners fall within the domain or change the Web provider to create "authenticated links" for any deep links from portlets to external or partner applications. Partner applications would have to be treated as external applications.
Moving provider UI servlet	Oracle9iAS Portal: Update location of the provider UI in General Settings.

Provider Groups (for Portals)

A provider group is a grouping of Portal Web providers. A provider group lets you publish information about many providers in a simple way. The provider group may be published from a portal, or it could be established independent of any portal. A portal administrator can register a provider group with a portal and view the contents (that is, the Web providers) of the group. The portal administrator can then selectively register one or more providers from the provider group with the portal and then use that provider as if it had been registered normally.

Table 7–6 Changes to Web Service Provider Groups

If you change this:	You affect these components:
Host, port, or mount point of SOAP engine	Oracle9iAS Portal: Update provider group registration information.
Service ID of provider groups service	
Location of group XML file	Oracle9iAS SOAP Service: Update service descriptors for the SOAP service.
Provider group name in progrp.xml	Oracle9iAS Portal: Update provider group registration information.

Wireless

You manage Oracle9iAS Wireless using Oracle Enterprise Manager Web site. The management pages enable you to perform tasks such as starting and stopping the wireless server, setting the number of connections in the connection pool, setting up logging, and configuring devices.

See the *Oracle9iAS Wireless Getting Started and System Guide* for details.

Table 7-7 Changes to Wireless

If you change this:	You affect these components:
Add a new service to refer to portal	Oracle9iAS Portal: Go to the Mobile tab of Global Settings and update the registered URL.

Oracle9iAS Single Sign-On

Oracle9iAS Single Sign-On enables users to sign in only once for all applications for which they are authorized to access. Oracle9iAS Single Sign-On is commonly used with portals, which can display multiple applications on a single page. Users enter their username and password only once to access the applications on the page.

You configure Oracle9iAS Single Sign-On using a Web-based interface. You can perform tasks such as setting the maximum session length, adding partner applications, and changing the administrator password.

See the *Oracle9iAS Single Sign-On Administrator's Guide* for details.

Table 7–8 Changes to Oracle9iAS Single Sign-On

If you change this:	You affect these components:
Hostname, Port, or Protocol (HTTP or HTTPS)	If you change the hostname, port, or protocol of Oracle HTTP Server, this affects Single Sign-On. See the section " Oracle HTTP Server " for details.

Oracle9iAS Portal:

Three cases are possible (for details on `ptlasst`, see the *Oracle9iAS Portal Configuration Guide*):

Case 1: If you are running Portal and Oracle9iAS Single Sign-On on the same machine, and you changed the name of the machine, you need to run the script:

(UNIX) `ORACLE_HOME/assistants/opca/ptlasst.csh`

(Windows) `ORACLE_HOME\assistants\opca\ptlasst.cmd`

in SSOPARTNERCONFIG mode:

```
ptlasst.csh -i typical -mode SSOPARTNERCONFIG -s portal -sp portal
-c webdbsvr2.us.oracle.com:1521:s901dev3 -sdad portal -o orasso
-odad orasso -host webdbsvr1.us.oracle.com -port 3000 -silent
-verbose -sso_c webdbsvr2.us.oracle.com:1521:s901dev3
-sso_h webdbsvr1.us.oracle.com -sso_p 3000 -pa orasso_pa
-pap orasso_pa -ps orasso_ps -pp orasso_ps -pd portal_dblink
-p_tns websso_ps -s_tns portal -iasname myIAS
```

Note: Running `ptlasst` in SSOPARTNERCONFIG mode creates a new row for the partner application instead of updating the existing row. This does not prevent Portal from working, but it can become a usability issue when the list of partner applications builds up on the Global Logout screen and some links might break.

(continued in next row)

Table 7-8 Changes to Oracle9iAS Single Sign-On (Cont.)

If you change this:	You affect these components:
(continued from previous row)	Oracle9iAS Portal (continued):
Hostname, Port, or Protocol (HTTP or HTTPS)	<p>Case 2: If you are running Portal and the Oracle9iAS Single Sign-On server on separate machines and you just changed the name of the Oracle9iAS Single Sign-On machine (the name of the Portal machine did not change), you need to run the script:</p> <pre>(UNIX) ORACLE_HOMEassistants/opca/ptlasst.csh (Windows) ORACLE_HOME\assistants\opca\ptlasst.cmd</pre> <p>in SSOPARTNERCONFIG mode. Specify the new Oracle9iAS Single Sign-On server name and port in the Oracle9iAS Single Sign-On server name and port parameters:</p> <pre>ptlasst.csh -i typical -mode SSOPARTNERCONFIG -s portal -sp portal -c webdbsvr2.us.oracle.com:1521:s901dev3 -sdad portal -o orasso -odad orasso -host webdbsvr1.us.oracle.com -port 3000 -silent -verbose -sso_c webdbsvr2.us.oracle.com:1521:s901dev3 -sso_h webdbsvr1.us.oracle.com -sso_p 3000 -pa orasso_pa -pap orasso_pa -ps orasso_ps -pp orasso_ps -pd portal_dblink -p_tns websso_ps -s_tns portal -iasname myIAS</pre> <p>Case 3: If you are pointing a Portal instance from one Oracle9iAS Single Sign-On server to another one, you need to do the following:</p> <ol style="list-style-type: none"> 1. Add the Portal as a partner application to the new Oracle9iAS Single Sign-On server. See the section "Adding a Partner Application" in Chapter 2 of the <i>Oracle9iAS Single Sign-On Administrator's Guide</i> for details. 2. Run the script: <pre>(UNIX) ORACLE_HOMEassistants/opca/ptlasst.csh (Windows) ORACLE_HOME\assistants\opca\ptlasst.cmd</pre> <p>in SSOPARTNERCONFIG mode to associate Portal with the new Oracle9iAS Single Sign-On server:</p> <pre>ptlasst.csh -i typical -mode SSOPARTNERCONFIG -s portal -sp portal -c webdbsvr2.us.oracle.com:1521:s901dev3 -sdad portal -o orasso -odad orasso -host webdbsvr1.us.oracle.com -port 3000 -silent -verbose -sso_c webdbsvr2.us.oracle.com:1521:s901dev3 -sso_h webdbsvr1.us.oracle.com -sso_p 3000 -pa orasso_pa -pap orasso_pa -ps orasso_ps -pp orasso_ps -pd portal_dblink -p_tns websso_ps -s_tns portal -iasname myIAS</pre> <p>Case 2 and Case 3 above assume that the Oracle Internet Directory server that the Oracle9iAS Single Sign-On servers point to is the same. If not, you also need to run <code>ssoconf.sql</code>. See the section "Oracle Internet Directory" for details.</p> <p>(continued in next row)</p>

Table 7–8 Changes to Oracle9iAS Single Sign-On (Cont.)

If you change this:	You affect these components:
(continued from previous row)	mod_osso:
Hostname, Port, or Protocol (HTTP or HTTPS)	<p>Run the Single Sign-On registration tool to re-register mod_osso. See the section "Reregistering the Oracle HTTP Server with the Single Sign-On Server" in the <i>Oracle9iAS Single Sign-On Release Notes</i> for details.</p> <p>Note: Running the registration tool creates a new row for the partner application instead of updating the existing row. This does not prevent the application from working, but it can become a usability issue when the list of partner applications builds up on the Global Logout screen and some links might break.</p>
	<p>Oracle9iAS Wireless:</p> <p>The registration tool for the Wireless partner application is:</p> <p>(UNIX) <code>ORACLE_HOME/wireless/sample/reRegisterSSO.sh</code></p> <p>(Windows) <code>ORACLE_HOME\wireless\sample\reRegisterSSO.bat</code></p> <p>Syntax:</p> <pre>reRegisterSSO.{sh,bat} <new_host_URL> <ORACLE_HOME></pre> <p>Example:</p> <pre>reRegisterSSO.sh http://new.myco.com:4567/ptg/rm /private/OraHome</pre> <p>Note: Running <code>reRegisterSSO</code> creates a new row for the partner application instead of updating the existing row. This does not prevent the application from working, but it can become a usability issue when the list of partner applications builds up on the Global Logout screen and some links might break.</p>
Password of ORASSO schema	Oracle HTTP Server and Oracle9iAS Portal:
You can change the password using the SSO Administration page or SQL*Plus.	<ol style="list-style-type: none"> 1. Use Oracle Enterprise Manager Web site to change the Database Password field for the DAD that accesses the Oracle9iAS Single Sign-On schema to match the new password. This changes the <code>PlsqlDatabasePassword</code> parameter in the <code>ORACLE_HOME/Apache/modplsql/conf/dads.conf</code> file. 2. Restart HTTP Server. <p>Oracle9iAS Portal:</p> <p>Update the <code>ssoServerPassword</code> property for the Oracle9iAS Single Sign-On target in the <code>ORACLE_HOME/sysman/emd/targets.xml</code> file.</p>

Table 7–8 Changes to Oracle9iAS Single Sign-On (Cont.)

If you change this:	You affect these components:
Password of ORASSO_PS schema	<p>Oracle9iAS Portal:</p> <p>No changes required if Portal uses the Metadata Repository API to retrieve the password. If Portal stores the password in a database table, then it has to update the table.</p>
Password of the lightweight SSO administrator (DN: cn=orcladmin, cn=users, o=company, dc=com)	No repercussions on other components.
Logout_url, success_url, failure_url, or home_url of partner applications, including mod_osso module	<p>Oracle9iAS Single Sign-On:</p> <p>You can change these URLs through the SSO Server Administration Page.</p>
Disable Oracle9iAS Single Sign-On	<p>Oracle9iAS Reports Services:</p> <p>Edit <code>ORACLE_HOME/reports/conf/rw servlet.properties</code> to set <code>SINGLESIGNON=NO</code> (default is <code>YES</code>).</p>
Any	<p>DAS:</p> <p>Changes to Oracle9iAS Single Sign-On are reflected by mod_osso, from which DAS gets Oracle9iAS Single Sign-On information.</p>
Oracle9iAS Single Sign-On application entry password, which is stored in Oracle Internet Directory at: orclApplicationCommonName =ORASSO, cn=SSO, cn=Products, cn=OracleContext You can change the password by changing the userPassword attribute using ODM or a command-line utility.	<p>Oracle9iAS Single Sign-On:</p> <p>Run <code>ORACLE_HOME/sso/admin/plsql/sso/ssooconf.sql</code> in SQL*Plus as the ORASSO user to update the password in the Oracle9iAS Single Sign-On preference store. This enables Oracle9iAS Single Sign-On to connect to Oracle Internet Directory.</p>
Load balancing	<p>Oracle HTTP Server and Load Balancer:</p> <p>If Oracle HTTP Server for the Oracle9iAS Single Sign-On server is behind a load balancer, you need to set the <code>KeepAlive</code> directive in the file <code>ORACLE_HOME/Apache/Apache/conf/httpd.conf</code> to <code>off</code>. If this directive is set to <code>on</code>, the load balancer maintains state with Oracle HTTP Server for the same connection, and this results in a HTTP 503 error code.</p>

Oracle Internet Directory

Oracle Internet Directory enables you to manage entities such as users and applications. You change configuration parameters for Oracle Internet Directory using Oracle Directory Manager or command-line tools such as `oidpasswd`.

See the *Oracle Internet Directory Administrator's Guide* for details.

Table 7–9 Changes to Oracle Internet Directory

If you change this:	You affect these components:
Oracle Internet Directory port, hostname, or application password	Components that depend on Oracle Internet Directory can pick up changes in Oracle Internet Directory's host and port if the components use the Repository API.
In the file <code>ORACLE_HOME/config/ias.properties</code> , change the value of <code>OIDhost</code> and <code>OIDport</code> accordingly.	<p>JAAS with LDAP:</p> <p>If you are using JAAS with LDAP, edit the <code>location</code> attribute in the <code>jazn</code> tag in the file <code>ORACLE_HOME/j2ee/home/config/jazn.xml</code>. The tag looks like the following:</p> <pre><jazn provider="LDAP" location="ldap://oidhost:port" /></pre> <p>If you specify the Oracle Internet Directory host and port as part of your application, then you have to edit your application deployment file.</p> <p>JAAS with XML:</p> <p>No changes necessary.</p> <p>Oracle9iAS Single Sign-On:</p> <p>If you move to a new Oracle Internet Directory instance, the new instance must be a replication of the old Oracle Internet Directory instance. In particular, the GUID of the users must be the same. Run <code>ORACLE_HOME/sso/admin/plsql/sso/ssooconf.sql</code> in SQL*Plus as the ORASSO user to enter the values for the new Oracle Internet Directory instance.</p>

Table 7–9 Changes to Oracle Internet Directory (Cont.)

If you change this:	You affect these components:
Portal application entry password, which is stored in Oracle Internet Directory at: orclApplicationCommonName=Portal, cn=Portal, cn=Products, cn=OracleContext	<p>Oracle9iAS Portal:</p> <p>Run <code>wwc/secupoid.sql</code> in the Portal schema. This updates the password in the Portal preference store, which allows Portal to connect to Oracle Internet Directory.</p> <p>Example of running <code>secupoid.sql</code> in SQL*Plus (you enter the values in bold):</p> <pre>SQL> @secupoid Current Configuration ----- OID Host: oid.domain.com OID Port: 389 Application DN: orclApplicationCommonName=PORTAL,cn=Portal, cn=Products,cn=OracleContext Application Password: 3E8C2D1B87CB61011757239C5AA9B390 Use SSL? N PL/SQL procedure successfully completed.</pre> <p>Updating OID Configuration Entries Press [Enter] to retain the current value for each parameter For SSL Connection to LDAP, specify "Y"es or "N)o</p> <pre>----- Enter value for oid_host: Enter value for oid_port: Enter value for app_password: mynewpassword Enter value for use_ssl_to_connect_to_ldap: Y Enter value for refresh_with_new_settings: N PL/SQL procedure successfully completed.</pre> <p>No errors.</p>
You can change the password by changing the <code>userPassword</code> attribute using ODM or a command-line utility.	

Table 7–9 Changes to Oracle Internet Directory (Cont.)

If you change this:	You affect these components:
Oracle9iAS Single Sign-On application entry password, which is stored in Oracle Internet Directory at: orclApplicationCommonName=ORASSO, cn=SSO, cn=Products, cn=OracleContext	Oracle9iAS Single Sign-On: Run <i>ORACLE_HOME/sso/admin/plsql/sso/ssooconf.sql</i> in SQL*Plus as the ORASSO user to update the password in the Oracle9iAS Single Sign-On preference store. This enables Oracle9iAS Single Sign-On to connect to Oracle Internet Directory.
You can change the password by changing the userPassword attribute using ODM or a command-line utility.	
Password for the cn=orcladmin account	JAAS: No effect.
You must use the <i>oidpasswd</i> utility to change the password; you cannot use any other tool.	Oracle9iAS Portal: No effect.
	Oracle9iAS Single Sign-On: No effect.

Metadata Repository

The metadata repository database contains schemas owned by various components. The database is installed when you install the Infrastructure for Oracle9i Application Server. Typically, you do not change anything in this database directly. Instead, you use provided interfaces, such as Oracle Enterprise Manager or utilities.

Table 7–10 Changes to the Metadata Repository

If you change this:	You affect these components:
Hostname for the repository database	Oracle9iAS Single Sign-On: Update the <code>ssoServerMachineName</code> property for the Oracle9iAS Single Sign-On target in the file <code>ORACLE_HOME/sysman/emd/targets.xml</code> . Note: If you have multiple repository database instances, Oracle9iAS Single Sign-On uses only one of the instances. You need to update the <code>ssoServerMachineName</code> property to point to the machine that hosts that instance.
Port for the repository database	Oracle9iAS Discoverer: Update the port for the <code>discodemo</code> service name in the <code>tnsnames.ora</code> file.

8

Deploying Applications

Oracle9iAS Containers for J2EE (OC4J) provides a complete set of Java 2 Enterprise Edition containers that allow you to deploy all types of J2EE applications. This chapter provides an introduction to deploying J2EE applications to OC4J using the Enterprise Manager Web site and uses the familiar Petstore demo as an example of deploying an application.

It contains the following topics:

- [Deploying J2EE Applications](#)
- [Deploying the Petstore Web Application Demo](#)
- [Recovering From Deployment Errors](#)
- [Undeploying J2EE Applications](#)

See Also: For complete information on deploying applications to OC4J, refer to *Oracle9iAS Containers for J2EE User's Guide*

Deploying J2EE Applications

You can deploy J2EE applications using the OC4J Home Page on the Enterprise Manager Web site. To navigate to an OC4J Home Page, do the following:

1. Navigate to the Instance Home Page where the OC4J instance resides. Scroll to the System Components section.
2. Select the OC4J instance in the Name column. This opens the OC4J Home Page for that OC4J instance.
3. Scroll to the Deployed Applications section on the OC4J Home Page.

[Figure 8-1](#) shows the Deployed Applications section. In this section, you can deploy applications using the **Deploy EAR file** or **Deploy WAR file** buttons. After deployment, you can modify the configuration of each application.

An application may require service configuration, such as data sources or security. This chapter does not cover such configuration. See *Oracle9iAS Containers for J2EE User's Guide* or *Oracle9iAS Containers for J2EE Services Guide* for more information. The [Deploying the Petstore Web Application Demo](#) section on page 8-10 provides an example of how you can configure data sources to support the application. In addition, the *Oracle9iAS Containers for J2EE User's Guide* provides more information on creating EAR and WAR files.

Figure 8-1 Deployed Applications Section

Deployed Applications

Default Application

Name [default](#)
Path [application.xml](#)

Applications

Select	Name	Path	Parent Application	Active Requests	Request Processing Time (secs)	Active EJB Methods
<input checked="" type="radio"/>	oc4jdemo	./applications/oc4jdemo.ear	default	0	0	0
<input type="radio"/>	petstore	./applications/petstore.ear	default	0	0	0
<input type="radio"/>	webappAdminDemo	./applications/webappAdminDemo.ear	default	0	0	0

Clicking **Deploy EAR File** or **Deploy WAR File** starts the deployment wizard, which deploys the application to the OC4J instance and binds any Web application to a URL context.

Note: You can also deploy simple applications with `dcmctl`, the DCM command-line utility. See [Appendix F, "DCM Command-Line Utility \(dcmctl\)"](#) for more information.

Basic Deployment

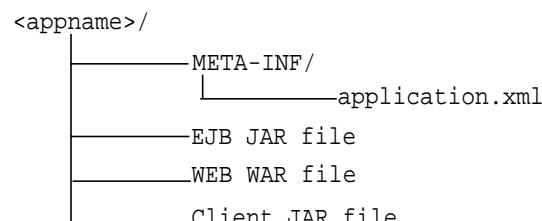
Your J2EE application can contain the following modules:

- Web applications
 - The Web applications module (WAR files) includes servlets and JSP pages.
- EJB applications
 - The EJB applications module (EJB JAR files) includes Enterprise JavaBeans (EJBs).
- Client application contained within a JAR file

Archive the JAR and WAR files that belong to an enterprise Java application into an EAR file for deployment to OC4J. The J2EE specifications define the layout for an EAR file.

The internal layout of an EAR file should be as follows:

Figure 8–2 Archive Directory Format



Archive these files using the JAR command in the <appname> directory, as follows:

```
% jar cvfM <appname>.EAR .
```

Note that the `application.xml` file acts as a manifest file.

- To deploy a J2EE application packaged within an EAR file, click **Deploy Ear File** in the Applications section of the OC4J Home Page.
- To deploy a J2EE Web application packaged within a WAR file, click **Deploy WAR File** in the Applications section of the OC4J Home Page.

Both of these buttons start an eight-step application deployment wizard that guides you through deploying an application. When you deploy an application within a WAR file, the `application.xml` file is created for the Web application. When you deploy an application within an EAR file, you must create the `application.xml` file within the EAR file. Thus, deploying a WAR file is an easier method for deploying a Web application.

Note: You must still provide configuration for J2EE services, such as data source and security configuration.

[Figure 8–3](#) shows the eight steps required for application deployment:

Figure 8–3 Deployment Wizard Steps



Introduction

The first page is an introduction to these steps. It reminds you to provide an EAR file with any OC4J-specific XML configuration files, if necessary. It also outlines some of the other steps in the deployment process.

Click **Next** to go to the next step in the wizard deployment process.

Select Application

[Figure 8–4](#) shows the second page, which enables you to browse your system for the EAR file to be deployed and provide a name to be identified with this application. The application name is user-created and will be the identifier for the application in the OC4J Home Page.

Figure 8–4 Designate Ear File**Deploy Application: Select Application**

Select the J2EE application (.ear file) to be deployed.

J2EE Application	<input type="text"/>	<input type="button" value="Browse..."/>
------------------	----------------------	--

Specify a unique application name for this application.

Application Name	<input type="text"/>
------------------	----------------------

When the application is deployed, the following occurs:

1. The EAR file is copied to the /applications directory.
2. A new entry is created in server.xml for the application, as follows:

```
<application name=<app_name> path=<path_EARfile> auto-start="true"
/>
```

where

- The name variable is the name of the application you provided.
- The path indicates the directory and filename where the EAR file is deployed.
- The auto-start variable indicates whether this application should be automatically restarted each time OC4J is restarted.

For a description of the elements in server.xml, see *Oracle9iAS Containers for J2EE User's Guide*.

Click **Next** to go to the next step in the wizard deployment process.

Provide The URL Mappings For All Web Modules

Map any Web modules in your application to a specific URL for its servlet context. All OC4J servlet contexts must be prefixed with a slash "/". When you try to access any Web applications, you provide the host, port, and Web context.

For all Web modules, your URL mapping for the module includes the URL you bind in this screen. Thus, for the URL `http://host:port/url_name`, provide `/url_name` in the URL mapping screen of the wizard.

Figure 8–5 URL Mappings for Web Modules**Deploy Application: URL Mapping for Web Modules**

A web module needs to be mapped to an URL pattern in the default web site before it can be accessed. The following table lists all the web modules found in your application. Specify the URL mapping for each of these modules.

Name	URL Binding
WebTier	*

Click **Next** to go to the next step in the wizard deployment process.

Provide Any Resource Reference Mappings

Map any resource references in your application, such as data sources or mail queues, to physical entities currently present on the OC4J container. Note that if you need a specific resource, you must have already added it to the OC4J container before you deploy your application in order for you to match them in this step.

For most applications, the resource reference you must designate is the data source JNDI name. This screen does not configure the data source information, it only designates an already configured data source or a data source that you will be configuring later. Designate the JNDI location name of the data source that the application uses.

Figure 8–6 Resource Reference Mappings**Deploy Application: Resource Reference Mappings**

The table below lists all resource references found in your application. Resource references need to be associated with the JNDI names of physical entities on the system where the selected instance/cluster is running.

Resource Reference	Type	Referenced By	Authentication	JNDI Location
jdbc/EstoreDataSource	javax.sql.DataSource	Web Module: WebTier	Container	
jdbc/EstoreDataSource	javax.sql.DataSource	EJB: TheProfileMgr	Container	
jdbc/InventoryDataSource	javax.sql.DataSource	EJB: TheInventory	Container	
jdbc/SignOnDataSource	javax.sql.DataSource	EJB: TheSignOn	Container	
jdbc/EstoreDataSource	javax.sql.DataSource	EJB: TheAccount	Container	
jdbc/EstoreDataSource	javax.sql.DataSource	EJB: TheOrder	Container	
mail/MailSession	javax.mail.Session	EJB: TheMailer	Container	
jdbc/EstoreDataSource	javax.sql.DataSource	EJB: TheCatalog	Container	

Click **Next** to go to the next step in the wizard deployment process.

Specify Any User Manager

You can specify what User Manager to use for security. For complete security, we recommend that you choose the JAZN XML User Manager.

Figure 8–7 User Manager Choices

User Manager

Specify a user manager to be associated with the application. Note that all web modules in your application will be automatically SSO enabled, when you use JAZN LDAP as your user manager.

<input checked="" type="radio"/> Use JAZN XML User Manager							
Default Realm <input type="text"/>							
XML Data File <input type="text"/> jazn-data.xml							
<input type="radio"/> Use XML User Manager							
Path to principals file <input type="text"/>							
<input type="radio"/> Use JAZN LDAP User Manager							
Default Realm <input type="text"/>							
LDAP Location <input type="text"/>							
<input type="radio"/> Use Custom User Manager							
Name <input type="text"/>							
Class Name <input type="text"/>							
Description <input type="text"/>							
Initialization Parameters for Class							
<table border="1"> <thead> <tr> <th>Select Name</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>No initialization parameters</td> <td></td> </tr> <tr> <td colspan="2">Add Another Row</td> </tr> </tbody> </table>		Select Name	Value	No initialization parameters		Add Another Row	
Select Name	Value						
No initialization parameters							
Add Another Row							

As [Figure 8–7](#) demonstrates, you must already have your User Manager set up and configured. Most of the entries require an XML file that designates the security roles, users, and groups for your security mappings.

The User Manager types are as follows:

- **JAZN XML User Manager**—This is the recommended User Manager. It requires a default realm and a `jazn-data.xml` file.
- **XML User Manager**—This is not the most secure option. It requires a `principal.xml` file.
- **JAZN LDAP User Manager**—This requires a default realm and an LDAP location.

- Custom User Manager—This User Manager must be programmed; provide the class name in this field.

For more information on security and User Managers, see both the *Oracle9iAS Containers for J2EE User's Guide* and the Security chapters in the *Oracle9iAS Containers for J2EE Services Guide*.

Provide Any Security Role Mappings

Map any security roles defined in your application to existing users and groups. If you have defined a security role within your application, you can map this role to a security group or role. You do not define security groups and users in this screen. Users and groups are obtained from your user manager.

Figure 8–8 Security Role Mappings

Deploy Application: Security Role Mappings

Your application exposes the following security roles. You may assign these roles to users/groups present on the OC4J container. To do this, select a role and then click on the Map Role button. You will be directed to a new page where you can map this role to users/groups. Click on OK in that page to get back to this screen and map another role.

Select	Name	Assigned Description	Assigned Groups
	No security roles found in this application		

Click **Next** to go to the next step in the wizard deployment process.

Publish Web Services

Publish any Web services defined in your application. This feature requires the UDDI registry. Web services are not installed with a core install.

If you have defined any Web services, they are displayed on the Publish Web Services Page.

Figure 8–9 Publish Web Services**Deploy Application: Publish Web Services**

The table below lists all of the web services found in your application. Each web service that you wish to access must be published to the UDDI registry in an appropriate category. To do this, select a web service and then click on the Publish button. You will be directed to a new page where you can enter details and select the category. Click on OK in that page to get back to this screen and publish another web service.

Web Services		
Select Web Service	Web Module	Status
No Web Services found		

If you want to publish these Web services, then click **Publish**. This leads you through the process of publishing your Web services. When finished, it brings you back to this screen.

Click **Next** to go to the next step in the wizard deployment process.

Summary of Deployment

At this point, you will receive a summary of your application deployment modules and configuration, as shown in [Figure 8–10](#).

Figure 8–10 Deploy Application Summary**Deploy Application: Summary**

Option	Value
Application	petstore.ear
Deployment Destination	Instance home1
Web Modules Mapped	1
Resource References Mapped	6
CMP Entity Bean Data Sources Mapped	5
Security Roles Mapped	0
User Manager	Use XML User Manager

Web Services

Web Services		
Web Service	Web Module	Status
No Web Services found		

TIP The HTTP listener will be restarted after deployment, to pick up the new web module mappings.

[Cancel](#) [Back](#) Step 8 of 8 [Deploy](#)

In order to deploy this application, click **Deploy**. The wizard displays a message that tells you that your application is deployed.

Post-Deployment Application Modifications

You can modify any fields and perform additional configuration tasks by returning to the OC4J Home Page and selecting the application name in the Applications section. This brings you to a screen with the details of the deployed application.

From within this screen, you can view the Web and EJB modules. In addition, you can add and modify application-specific properties, resources, and security options in the Administration section. It is in this Administration section that you can add application-specific data sources or security groups or users mentioned in the deployment wizard.

Deploying the Petstore Web Application Demo

This section uses the Petstore demo as an example of how to configure and deploy a J2EE application. All OC4J server configuration and modifications to the Petstore application configuration have been performed for you. You can execute the Petstore demo with minimal effort to see how OC4J works.

Note: Displays of the screens for each step of the deployment wizard are shown in "[Deploying J2EE Applications](#)" on page 8-2.

Downloading an OC4J-Ready Petstore Demo

This section contains instructions for downloading and deploying an OC4J-ready version of the Petstore demo. You must have a working Oracle database and an OC4J installation. You should use this installation for demonstration purposes only and not in a production environment. In this simplified version, we have pre-built the Petstore demo using Oracle database (instead of the default Cloudscape) and edited the configuration files to make the setup easy.

The Petstore demo is available on Oracle Technology Network (OTN). To install, configure, and run the Petstore demo:

1. Navigate to

http://otn.oracle.com/sample_code/tech/java/oc4j/content.html

and find the Petstore Demo in the Sample Code section.

2. Download the `jps112.zip` file, which contains version 1.1.2 of the Petstore demo. This ZIP file contains an annotated version of this application, along with preconfigured OC4J XML files.
3. Unzip the `jps112.zip` file, which contains the following:
 - a. `steps.html`—Steps on how to deploy the Petstore demo in standalone mode. This HTML file does not contain directions on how to deploy using the Oracle Enterprise Manager. The steps in this book instruct you on how to deploy using Enterprise Manager.
 - b. `petstore.ear`—The Petstore demo is contained in `petstore.ear`.
 - c. `config.zip`—OC4J server XML configuration files are provided for you in `config.zip` file.
4. Unzip the `config.zip` file to retrieve the `server.xml`, `default-web-site.xml`, and `data-sources.xml` files.
5. Edit `data-sources.xml` to point to your database by replacing the host, port, and sid in the `url` attribute for the database as follows:

```
url="jdbc:oracle:thin:@host:port:sid"
```
6. Create the user `estoreuser` in your database and grant this user privileges to connect as `SYSDBA` to your database. You can create the user and grant privileges through the following command:

```
SQL> grant connect, resource to estoreuser identified by estore;
```
7. Navigate to the OC4J Home Page on the Oracle Enterprise Manager Web site.
8. Select **default** under the Default Application section. The default application is the automatic parent of each application and it holds global configuration information for all deployed applications, such as the data sources used. You are going to add the data sources that Petstore uses in the default application.
9. Add data sources. On the Default Application Page, scroll to the Administration section and select **Advanced Properties** in the Properties column.

Since the data sources are provided in a `data-sources.xml` file, add these data sources using the XML editor within Enterprise Manager. Select `data-sources.xml` in the filename column. This brings up a screen with XML in a text window. Merge the data sources from the `data-sources.xml` that was provided within the `config.zip` of the Petstore download into the file in this window. Do not overwrite other data sources already configured in this file. When finished, click **Apply**.

Note: Because you were provided the `data-sources.xml` file, you can add or modify this file directly through Advanced Properties. If you do not have the XML file, you can add the configuration details through the Data Sources option under the Resources column.

10. Return to the OC4J Home Page and scroll to the Applications section. Click **Deploy EAR File**. This starts the application deployment wizard.
11. Read the introduction to the deployment wizard. Click **Next**.
12. Provide the EAR file and the name of your application on the Select Application Page. Click **Browse** to find the `petstore.ear` file that you downloaded to your system. Type "petstore" in the application name field. Click **Next**.
13. Provide the URL mappings for the servlet context on all Web modules in the Petstore application. The Petstore demo contains a single Web module, which should be mapped to the `/estore` servlet context. Type `"/estore"` in the URL mapping field and click **Next**.
14. At this point, the Petstore demo does not need any additional configuration through the wizard. Click **Finish**, which takes you to the Summary Page.
15. Read the summary of the Petstore application deployment. Click **Deploy** to complete the application deployment.
16. Select "petstore" in the Name column of the Applications section on the OC4J Home Page. This shows the configuration and all deployed modules of the Petstore demo application. If the OC4J server is started, the application is automatically started.
17. Execute the Petstore application in your browser by accessing the Oracle HTTP Server, where the default port is 7777.

`http://http_server_host:http_server_port/estore`

The Petstore splash screen appears. Follow the instructions provided by the Petstore application to load the Java Petstore database tables.

Recovering From Deployment Errors

If the deployment process is interrupted for any reason, you may need to clean up the temp directory, which by default is:

- (UNIX) /var/tmp
- (Windows) The directory defined by the TEMP system variable

The deployment wizard uses 20 MB of swap space in the temp directory for storing information during the deployment process. At completion, the deployment wizard cleans up the temp directory by removing its files. However, if the wizard is interrupted, it may not have the time or opportunity to clean up the temp directory. Thus, you must clean up any additional deployment files from this directory yourself. If you do not, this directory may fill up, which will disable any further deployment. If you receive an Out of Memory error, check for space available in the temp directory.

If the temp directory does not have enough space, you can configure OC4J to use a different temp directory using the Enterprise Manager Web site:

1. Navigate to the OC4J Home Page. Scroll to the Administration Section.
2. Select **Server Properties** in the Administration section. This opens the Server Properties Page.
3. Scroll to the Command Line Options section. Add the following variable definition to the OC4J Options line:

`java.io.tmpdir=new_temp_dir`

where *new_temp_dir* is the full path to the new temporary directory.

All new OC4J processes will use the new temp directory.

Undeploying J2EE Applications

You can undeploy a J2EE Web application using the Enterprise Manager Web site:

1. Navigate to the OC4J Home Page that contains the application you would like to undeploy. Scroll to the Applications section.
2. Click the radio button in the Select column for the application.

3. Click Undeploy.

This operation results in the following:

- The application is removed from the OC4J runtime.
- All bindings for the Web modules are removed from all the Web sites to which the Web modules were bound.
- The application files are removed from both the applications and application-deployments directories.

Note: You can also undeploy applications with `dcmctl`, the DCM command-line utility. See [Appendix F, "DCM Command-Line Utility \(dcmctl\)"](#) for directions.

Monitoring HTTP Server, OC4J, and Web Cache

This chapter provides an introduction to monitoring application server instances and components.

It contains the following topics:

- [Introduction](#)
- [Monitoring Oracle HTTP Server](#)
- [Monitoring Oracle9iAS Containers for J2EE](#)
- [Monitoring Oracle9iAS Web Cache](#)

Introduction

This chapter provides an introduction to monitoring components using the Oracle Enterprise Manager Web site. It discusses the available metrics for three key components: Oracle HTTP Server, Oracle9iAS Containers for J2EE, and Oracle9iAS Web Cache.

See Also: Enterprise Manager Web site allows you to monitor many other Oracle9iAS components. For more information, refer to the individual component guides and use the Enterprise Manager Web site online help.

Monitoring Oracle HTTP Server

You can use the Enterprise Manager Web site to monitor the activity of Oracle HTTP Server. The following sections describe the HTTP Server Home Page and related pages, which allow you to monitor the server status, server modules, and the server load and response metrics.

Oracle HTTP Server Home Page

The HTTP Server Home Page provides you with a high-level overview of the HTTP Server. You can access the HTTP Server Home Page in the Name column of the System Components table on the Instance Home Page. [Figure 9-1](#) displays the HTTP Server Home Page. This is where you begin your HTTP Server monitoring tasks:

- The General section shows you the current status of the server.
- The Default Server Configuration section provides read-only information on the current configuration settings.
- The Status section provides an at-a-glance view of the overall stability of your server.
- The Response and Load section provides at-a-glance details about the performance of the server.

Note: The performance metrics collected for virtual hosts are categorized by virtual host name only. If your site deploys multiple virtual hosts that differ only in their listening port or IP address, their performance metrics will be displayed summed together.

Figure 9–1 HTTP Server Home Page

The screenshot shows the Oracle Enterprise Manager interface for Application Servers. The title bar says "Enterprise Manager". The top menu has "Targets" selected. The main navigation bar shows "Farm > Application Server: ias1.myserver.mycompany.com > HTTP Server". Below this is a sub-navigation bar with "HTTP Server". The page content is divided into sections: "General", "Default Server Configuration", "Status", and "Response and Load".

General

Status	Up	Stop	Restart
	Mar 7, 2002		
Start Time	3:27:20 PM		
	PST		

Default Server Configuration

Server Name	myserver.mycompany.com
Document Root	/private/oracle/OraHome1
Last Modification	Mar 6, 2002 11:20:32 AM PST

Status

Heap Usage (MB)	17.852
CPU Usage (%)	0.1
Memory Usage (MB)	261.248
Error Rate (%)	0
Active Connections	2
Connection Open Time (sec)	0.02

Response and Load

Active Requests	1
Request Throughput (reqs/sec)	0.43
Request Processing Time (sec)	0.02
Data Throughput (KB/sec)	8.1
Data Processed (KB/req)	18.72

To monitor the HTTP Server from a more detailed point of view, use the Performance section of the HTTP Server Home Page to access the following metric pages:

- [Status Metrics Page](#)
- [Module Metrics Page](#)
- [Monitoring Oracle9iAS Containers for J2EE](#)

Status Metrics Page

You can view detailed status information about your HTTP Server on the Status Metrics Page on the Oracle Enterprise Manager. [Table 9–1](#) shows you what to look for in each section of the page.

Table 9–1 Using the Status Metrics Page

Section	Things to Look For
System Usage Metrics	Review the CPU Usage and Memory Usage charts for a snapshot of the server's overall system usage. Specifically, monitor the system for CPU bounding, which occurs when the CPU utilization is around 90%.

Table 9–1 Using the Status Metrics Page (Cont.)

Section	Things to Look For
Error Metrics	<p>Review the overall number of error conditions that exist in the server. For example, check the Error Rate Since Startup metric to see if there is an increase in errors over time. The Errors by Error Type table identifies the types of errors users are receiving from the server.</p> <p>Select Error Log in the Related Links section to review the error log file, which can help you troubleshoot problems with your server. Select Logging Properties to configure the logging properties of the server so that it can log more detailed information for analysis.</p>
Connection Metrics	<p>Review the number of Active Connections and the Connections Completed Since Startup to see the level of activity on this server. If you have a high number of connections, check the average time taken for each connection. For example, if you are supporting a high number of connections and have a low average connection time, consider modifying the logic of your application so it requires fewer individual connections.</p>

To view status information:

1. Navigate to the HTTP Server Home Page on the Oracle Enterprise Manager. Scroll to the Performance section.
2. Select **Status Metrics**. This opens the Status Metrics Page.
3. Review the metrics as listed in [Table 9–1](#).
4. Click **Help** at the top of the page for more information about the individual metrics.

Module Metrics Page

You can determine the performance of individual Oracle HTTP Server modules using the Module Metrics Page on the Oracle Enterprise Manager.

HTTP Server modules extend your system by allowing you to add features, develop and test features independently, and enable or disable features as needed.

Note: To use a module, you must first register it with Oracle HTTP Server. This is done by modifying the `http.config.h` file. For more information, see *Oracle HTTP Server Administration Guide*.

To view metrics for your Oracle HTTP Server modules:

1. Navigate to the HTTP Server Home Page on the Oracle Enterprise Manager Web site. Scroll to the Performance section.
2. Select **Module Metrics**. This opens the Module Metrics Page.

The Module Metrics page displays four request metrics for each Oracle HTTP Server module. If you detect poor performance of a particular module, see the error log to determine the cause.

3. Click **Help** at the top of the page for more information about the individual metrics.

Response and Load Metrics Page

You can view an overall snapshot of how the server is performing on the Response and Load Metrics Page on the Oracle Enterprise Manager.

1. Navigate to the HTTP Server Home Page on the Oracle Enterprise Manager Web site. Scroll to the Performance section.

2. Select **Response and Load Metrics**. This opens the Response and Load Metrics Page.

Response metrics indicate (on average) how many requests were submitted and how long it took for the server to respond to a user's request. The Load metrics indicate (on average) how many bytes of data the server processed with the requests.

You should expect response times to be slower when larger loads are being processed, but you should also account for heavy usage in your server's initial configuration. See the *Oracle9i Application Server Performance Guide* for more information on handling response and load time.

3. Click **Help** at the top of the page for more information about the specific metrics.

Monitoring Oracle9iAS Containers for J2EE

You can use the Enterprise Manager Web site to view the running status of OC4J instances and monitor the availability, usage, and overall performance of the OC4J server and its applications.

OC4J Home Page

The OC4J Home Page is the starting point for monitoring OC4J. You can access the OC4J Home Page by selecting the OC4J instance in the Name column of the System Components table on the Instance Home Page. [Figure 9–2](#) displays the OC4J Home Page, which displays usage and performance data for OC4J applications. Using the applications table, you can determine which application is incurring the highest volume of requests or longest response time. You can examine an application further by drilling down into the application to view the performance of individual objects, such as servlets and EJBs.

Figure 9–2 OC4J Home Page

The screenshot shows the Oracle Enterprise Manager Application Servers home page for an OC4J instance. At the top, there's a navigation bar with links for Preferences and Help, and a Targets button. Below that, a breadcrumb trail shows Farm > Application Server: ias1.myserver.mycompany.com > home. The main content area is titled "home" and includes a timestamp: Refreshed at Friday, March 8, 2002 8:23:19 AM PST.

General

Status	Up Mar 7, 2002 3:27:00 PM PST
Start Time	3:27:00 PM PST
Virtual Machines	1

Status

CPU Usage (%)	5.4
Memory Usage (MB)	109.424
Heap Usage (MB)	13.984

JDBC Usage

Open JDBC Connections	8
Total JDBC Connections	1,516
Active Transactions	5
Transaction Commits	182
Transaction Rollbacks	69

Response - Servlets and JSPs

Active Sessions	38
Active Requests	1
Request Processing Time (secs)	0.022
Requests per Second	0.223

Response - EJBs

Active EJB Methods	25
Method Execution Rate (per sec)	47

The OC4J Home Page displays a single view of the instance and presents a roll up of status and performance metrics, including:

- How long the instance has been up
- OC4J system resource usage, such as the percentage of CPU and memory resources being consumed
- Volume and average processing time of application requests and transactions

The OC4J Home Page is divided into several sections that cover server status and metrics, data about deployed applications, and property and default settings information.

The **General** section displays the current running status of the selected OC4J instance. This section includes the Status, Start Time, and number of virtual machines used to execute OC4J application code for the current OC4J instance. You can choose to stop or restart the server from this section.

The **Status** section displays the key metrics listed in [Table 9–2](#) for monitoring the overall OC4J instance health. In general, these metrics focus on three factors: volumes, rates, and duration.

Table 9–2 Status Section on OC4J Home Page

Metric	Description
CPU Usage (%)	Percentage of the CPU being used for the OC4J instance. As usage increases, the CPU utilization increases.
Memory Usage (MB)	Number of megabytes being utilized for OC4J memory.
Heap Usage (MB)	Total amount of heap space used (in megabytes) since the server was started.

The **JDBC Usage** section provides information about the JDBC connections and transactions and includes the metrics listed in [Table 9–3](#).

Table 9–3 JDBC Usage Section on OC4J Home Page

Metric	Description
Open JDBC Connections	Number of open JDBC connections.
Total JDBC Connections	Number of total JDBC connections.
Active Transactions	Number of transactions that are currently open from Enterprise Java Beans in EJB modules.
Transaction Commits	Number of Transactions committed by Enterprise Java Beans in EJB modules.
Transaction Rollbacks	Number of times a transaction was rolled back by Enterprise Java Beans in EJB modules.

The **Response - Servlets and JSPs** section provides the transactional details about the server and includes the metrics listed in [Table 9–4](#).

Table 9–4 Response - Servlets and JSPs section on OC4J Home Page

Metric	Description
Active Sessions	Number of currently active sessions for the specified OC4J instance
Active Requests	Total number of active requests currently being processed
Request Processing Time (secs)	Amount of time in seconds spent processing requests during the last period
Requests per Second	Number of requests processed during the last period divided by the time interval of the period

The **Response - EJBs** section provides transactional details about the Enterprise Java Beans and includes the metrics seen in [Table 9–5](#).

Table 9–5 Response - EJBs section on OC4J Home Page

Metric	Description
Active EJB Methods	Number of EJB Methods currently active
Method Execution Rate (per sec)	Rate of completion of the EJB Client Methods

The **Deployed Applications** section displays a list of all OC4J applications deployed in the OC4J instance, including the status of each application measured by the number of active requests and average processing time for the application. You can perform the following tasks in this section:

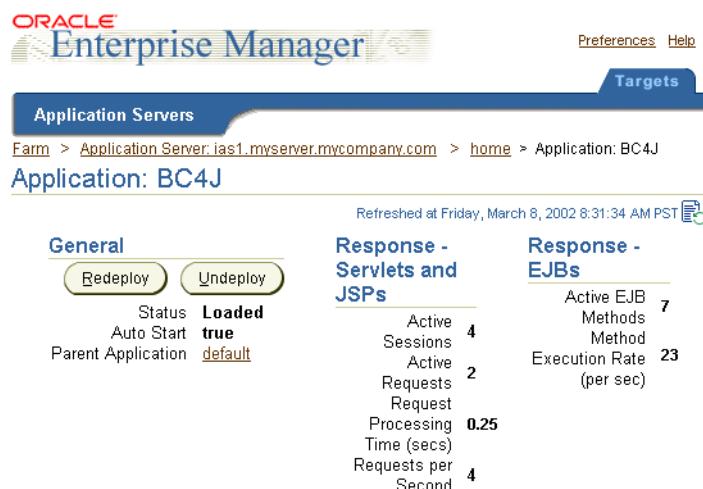
- Deploy a new OC4J application
- Redeploy or undeploy an OC4J application

The **Applications table** displays a list of all installed applications in the OC4J instance. You can select any application to display the Application Page for that application, which includes more performance statistics for the specified application.

The **Administration** section enables you to set instance properties, such as server and Web site properties, and application defaults, such as data sources and principals. The Administration section also includes related links to the UDDI registry and JAAS Policy Management.

Application Pages

The OC4J Application Page displays information and metrics about the specified application. You can access the Application Page by selecting the name of the application in the Name column of the Application table on the OC4J Home Page.

Figure 9–3 OC4J Application Page

The Application Page provides application status information along with response metrics for Servlets, JSPs, and EJBs. There are two tables in the center of the page that list Web Modules and EJB Modules.

The **General** section shows the current status of the selected OC4J application and includes a Response section for Servlets, JSPs, and EJBs. You can redeploy or undeploy an OC4J application from the General section.

The **Response - Servlets and JSPs** section provides transactional details about the server and includes the metrics seen in [Table 9–6](#).

Table 9–6 Response - Servlets and JSPs Section on OC4J Applications Page

Metric	Description
Active Sessions	Number of currently active sessions for the specified OC4J application
Active Requests	Total number of active requests currently being processed
Request Processing Time (secs)	Amount of time in seconds spent processing requests during the last period
Requests per Second	Number of requests processed during the last period divided by the time interval of the period

The **Response - EJBs** section provides transactional details about the Enterprise Java Beans and includes the metrics seen in [Table 9–7](#).

Table 9–7 Response - EJBs section on OC4J Application Page

Metric	Description
Active EJB Methods	Number of EJB Methods currently active
Method Execution Rate (per sec)	Rate of completion of the EJB Client Methods

The **Web Modules** table displays a list of all Web modules used in the OC4J application. You can select a Web module in the Name column to drill down to the Overview Page for the Web module. The Web Module table includes the Web module name, the path, the number of active requests, the request processing time, and the number of active sessions.

The **EJB Modules** table displays a list of all Enterprise Java Beans modules used in the OC4J application. You can select an EJB module in the Name column to drill down to the Overview Page for that specific EJB module. The EJB Modules table lists the EJB Module name, the path, the number of EJBs deployed, the number of active EJB methods, and the method execution rate per second for each EJB module.

The **Administration** section provides access to basic and advanced configuration functions, data sources and their settings, and security principals and roles.

You can use the Application Page to diagnose and isolate problems involving specific servlets or JSPs. For example, if your Web application is experiencing user delays, you can use the OC4J Application Page to diagnose and isolate the servlets or JSPs responsible for the longest response times. You can use the Overview for Web Module Page or the EJB Module Page to break down further the performance of servlets, JSPs, or EJBs, looking at load times, service times, and overhead.

Monitoring Oracle9iAS Web Cache

Oracle9iAS Web Cache is a content-aware server accelerator, or reverse proxy server, that improves the performance, scalability, and availability of Web sites that run on Oracle9i Application Server.

By storing frequently accessed URLs in memory, Oracle9iAS Web Cache eliminates the need to repeatedly process requests for those URLs on the application Web server.

Unlike legacy proxy servers that handle only static documents, Oracle9iAS Web Cache caches both static and dynamically generated content from one or more application Web servers. Because Oracle9iAS Web Cache can cache more content than legacy proxies, it provides optimal performance by greatly reducing the load on application Web servers.

Oracle9iAS Web Cache sits in front of application Web servers, caching their content and providing that content to Web browsers that request it. When Web browsers access the Web site, they send HTTP protocol or HTTPS protocol requests to Oracle9iAS Web Cache. Oracle9iAS Web Cache, in turn, acts as a virtual server to the application Web servers. If the requested content has changed, Oracle9iAS Web Cache retrieves the new content from the application Web servers. The application Web servers may retrieve their content from an Oracle database.

Oracle9iAS Web Cache Home Page

The Web Cache Home Page enables you to monitor the running status of Oracle9iAS Web Cache, as well as the overall performance statistics for Oracle9iAS Web Cache. You can access the Web Cache Home Page by selecting an Oracle9iAS Web Cache server in the Name column of the System Configuration table on the Oracle9iAS Instance Home Page. [Figure 9–4](#) displays the Web Cache Home Page.

Figure 9–4 Web Cache Home Page



The **General** section displays the current running status of Oracle9iAS Web Cache. It also displays the key metrics listed in [Table 9–8](#) for monitoring overall Oracle9iAS Web Cache health.

Table 9–8 General Section in Web Cache Home Page

Metric	Description
CPU Usage (%)	Percentage of the CPU being used for Oracle9iAS Web Cache (As traffic increases, the CPU utilization increases.)
Memory Usage (MB)	Number of megabytes being utilized for cache memory
Number of Cached Documents	Number of documents currently cached
Size of Cached Documents (MB)	Number of megabytes utilized by documents in the cache (MB)

The **Activity** section displays the cache resource and performance information listed in [Table 9–9](#).

Table 9–9 Activity Section in Web Cache Home Page

Metric	Description
Requests Per Second	Number of requests that Oracle9iAS Web Cache is currently serving each second
Data Served (MB/sec)	Number of megabytes that Oracle9iAS Web Cache is currently serving each second
Apologies Pages Served (% of requests)	Percentage of apology pages that Oracle9iAS Web Cache is serving to Web browsers, due to a network or busy Web site error
Cache Cluster	Name of the cache cluster. This field displays only if a cache cluster is configured.
Open Connections	Current number of incoming open connections to the Oracle9iAS Web Cache server and outgoing open connections to the application Web servers

The Performance section provides access to the following pages:

- [Activity Page](#)
- [Application Web Servers Page](#)
- [Popular Cached Documents Page](#)

Activity Page

The Web Cache Activity Page enables you to monitor the efficiency of the cache. It displays metrics for cache hit requests, compression requests, and apology page requests.

To access the Activity Page:

1. Navigate to the Web Cache Home Page on the Oracle Enterprise Manager Web Site.
2. Scroll to the Performance section.
3. Select **Activity**. This opens the Web Cache Activity Page.

Figure 9–5 displays the Web Cache Activity Page.

Figure 9–5 Web Cache Activity Page

The screenshot shows the Oracle Enterprise Manager interface for monitoring Application Servers. The main title is "Enterprise Manager". The navigation bar includes "Targets" (selected), "Preferences", and "Help". The main content area is titled "Activity". The "Requests" section displays current per-second metrics: Current per Second (154), Average per Second (114), Maximum per Second (168), and Total Requests (1,080,831). The "Hit Rate" section shows a table of hits and misses with their respective counts and percentages. The "Compression" section provides statistics for compressed hits, misses, served bytes, and saved bytes. The "Apology Pages Served" section shows a table for network errors, site busy errors, and partial page errors. At the bottom, there are links for "Targets", "Preferences", and "Help", and a copyright notice: "Copyright © 1996, 2002, Oracle Corporation. All rights reserved."

	Current (per second)	Current Requests (%)	Total
Compressed Hits	105	68	740,135
Compressed Misses	0	68	160
Served (MB)	0.344		2,414.884
Saved by Compression (MB)	0.288		2,023.115

	Current (per second)	Current Requests (%)	Total
Network Errors	0	0	0
Site Busy Errors	0	0	0
Partial Page Errors	0	0	0

For cache hit rates, compression, and apology pages, the page provides the following information:

- Documents currently being served by Oracle9iAS Web Cache each second in the **Current (per second)** columns
- Percentage of documents currently being served by Oracle9iAS Web Cache each second in the **Current Requests (%)** columns
- Total documents served since Oracle9iAS Web Cache was started or restarted in the **Total** columns

Application Web Servers Page

The Web Cache Application Web Servers Page enables you to view the status of the application Web servers to which Oracle9iAS Web Cache forwards requests.

To access the Web Cache Activity Page:

1. Navigate to the Web Cache Home Page on the Oracle Enterprise Manager Web Site.
2. Scroll to the Performance section.
3. Select **Application Web Servers**. This opens the Application Web Servers Page.

[Figure 9–6](#) displays the Web Cache Application Web Servers Page.

Figure 9–6 Web Cache Application Web Servers Page

The screenshot shows the Oracle Enterprise Manager interface with the title "Enterprise Manager". The "Targets" tab is selected. Below it, the "Application Servers" tab is selected. The URL in the address bar is "Application Server: IAS-1 > IAS-1 WebCache-myserver.mycompany.com > Application Web Servers". The page is refreshed at "Friday, January 4, 2002 2:48:07 PM EST".

Name	Up/Down	Proxy Server	Active Requests	Current Requests Per Second	Average Requests Per Second	Current Processing Time (sec)	Average Processing Time (sec)
myserver:7777	✓	No	0	14	36	0	1
myserver:4444	✓	No	0	0	0	0	0

At the bottom, there are links for "Targets", "Preferences", and "Help", and a copyright notice: "Copyright © 1996, 2002, Oracle Corporation. All rights reserved."

View the Up/Down column. An Up status specifies that the last communication with the server was successful. A Down status specifies that the server is down. If this is the last server in a single or multiple server configuration, Oracle9iAS Web Cache continues to forward requests to the server. If this is not the last server, then no new requests are sent to server. However, Oracle9iAS Web Cache polls the inactive server until it is back online.

To view detailed metrics for a specific application Web server:

1. Select an application Web server host name to view. This opens the Application Web Server Page.
2. View the metrics in which you are interested:
 - The Active Requests metric in the Requests Throughput section shows the current load
 - The metrics in the Request Processing Time section shows processing latency
 - The Concurrent Bound Sessions metric in the Connections section shows the number of open connections

Figure 9–7 displays the Web Cache Application Web Server Page for myserver.mycompany.com:4444.

Figure 9–7 Web Cache Application Web Server Page

The screenshot shows the Oracle Enterprise Manager interface for monitoring application web servers. The title bar reads "Enterprise Manager". The main navigation bar includes "Targets" (which is selected), "Preferences", and "Help". Below the navigation, the path is shown as "Application Server: IAS-1 > IAS-1_WebCache-myserver.mycompany.com > Application Web Servers > Application Web Server: myserver.mycompany.com".

Status: Up
 Web Cache Start Time: Jan 4, 2002 2:12:32 PM
 Proxy Server: No

Refreshed at Friday, January 4, 2002 2:49:03 PM EST

Request Throughput		Request Processing Time		Connections	
Active Requests	1	Current Processing Time (sec)	0	Concurrent Bound Sessions	0
Current Requests Per Second	17	Average Processing Time (sec)	1	Maximum Bound Sessions	0
Average Requests Per Second	36				
Maximum Requests Per Second	55				
Maximum Number of Concurrent Requests	1				
Total Requests	5,937				

Targets | Preferences | Help
 Copyright © 1996, 2002, Oracle Corporation. All rights reserved.

Popular Cached Documents Page

The Web Cache Popular Cached Documents Page enables you to view the URLs of the most popular documents in the cache and to generate a list of the URLs of all the objects currently in the cache. This information helps you verify that the cacheability rules are configured correctly: that the objects that you want cached are being cached. [Figure 9–8](#) displays the Web Cache Popular Cached Documents Page.

Figure 9–8 Web Cache Popular Cached Documents Page

The screenshot shows the Oracle Enterprise Manager Application Servers interface. The title bar reads "ORACLE Enterprise Manager". The top navigation bar includes "Targets" (which is selected), "Preferences", and "Help". Below the title bar, a breadcrumb trail shows "Application Server: IAS-1 > IAS-1_WebCache-myserver.mycompany.com > Popular Cached Documents". The main content area is titled "Popular Cached Documents" and includes a timestamp "Refreshed at Friday, January 4, 2002 2:47:33 PM EST". A note states: "The list below shows the popularity ranking of documents in the cache. The ranking is based on how frequently and how recently the document was read." A warning message follows: "Warning: The performance of the cache is significantly impacted when generating this list. While the list is being generated, other requests to the cache are blocked and users will have to wait for an extended period of time for their requests. The more documents there are in the cache, the greater the performance impact. To minimize the performance impact, you should show this list only during periods of low activity." Below this, it says "Number of Cached Documents 5". There is a "Show Popular Documents" button and a pagination control "1-5 of 5". A table lists the top 5 cached documents:

Rank	URL	Size (bytes)	Rule Regular Expression *	Rule Site Mask *
1	/myserver.mycompany.com:7787/new.htm ;gzip	912	\.html?\$/	None
2	/myserver.mycompany.com:7787/tour.htm ;gzip	964	\.html?\$/	None
3	/myserver.mycompany.com:7787/demos.htm ;gzip	1.23 K	\.html?\$/	None
4	/myserver.mycompany.com:7787/docs.htm ;gzip	943	\.html?\$/	None
5	/myserver.mycompany.com:7787/manual/index.html ;gzip	2.06 K	\.html?\$/	None

* A document is cached if its URL matches the regular expression of a caching rule or if caching is specified in the document's http response header. A document cached because of its header has no associated rule.

Targets | Preferences | Help
Copyright ©1996, 2002, Oracle Corporation. All rights reserved.

Note: The Popular Cached Documents Page lists only those documents that are valid. Although the cache may contain documents that are expired or that have been invalidated, those documents are not included in the list of popular documents.

To access the Popular Cached Documents Page:

1. Navigate to the Web Cache Home Page on the Oracle Enterprise Manager Web Site.
2. Scroll to the Performance section.
3. Select **Popular Cached Documents**. This opens the Popular Cached Documents Page.

Reconfiguring the Application Server

This chapter describes tasks for reconfiguring an application server enterprise.

It contains the following topics:

- [Configuring Additional Components](#)
- [Extending an Application Server Installation](#)
- [Adding an Additional Infrastructure](#)
- [Associating an Instance with an Infrastructure \(Joining a Farm\)](#)
- [Changing the Schema Used by a Component](#)
- [Changing Component Schema Passwords](#)

Configuring Additional Components

You can add functionality to your application server by configuring components that were not configured during installation.

Note: If you intend to configure JServ, refer to "[Enabling JServ](#)" on page 6-12 before following the steps in this section.

You can configure components as follows using the Enterprise Manager Web site:

1. Start a browser from the host that the component you would like to configure resides on. You must do this because some components launch a GUI when you configure them.
2. Navigate to the Instance Home Page on the Enterprise Manager Web site. Scroll to the Administration section.
3. Select **Configure Components**. This opens the Configure Components Page.
4. Choose the component you would like to configure from the drop-down menu, enter the `ias_admin` password, and click **OK**.

If you configure any of the following components, you may need to take additional steps to setup the component to use a customer database:

- Oracle9iAS Clickstream Intelligence
- Oracle9iAS Discoverer
- Oracle9iAS Personalization
- Oracle9iAS Portal
- Oracle9iAS Unified Messaging

See Also: ["Task 3: Complete Component Configuration"](#) on page 4-4 for more details

Once you configure a component, it cannot be deconfigured, but you can remove its functionality by not starting it when you start the rest of the components in your installation.

Extending an Application Server Installation

You can extend an application server installation by installing a higher install type in the same Oracle home. For example, you can extend a Portal and Wireless installation to a Business Intelligence and Forms installation or a Unified Messaging installation. To extend an application server installation, run Oracle Universal Installer using the Oracle home you would like to extend.

Note: You cannot reduce an application server installation by installing a lower install type in the same Oracle home. For example, you cannot install a J2EE and Web Cache installation in an Oracle home that contains a Portal and Wireless installation.

See Also: *Oracle9i Application Server Installation Guide* for more information

Adding an Additional Infrastructure

Most application server enterprises have one infrastructure that contains Oracle9iAS Single Sign-On, Oracle Internet Directory, a metadata repository, and, optionally, Oracle Management Server. If the load on your metadata repository is at full capacity, you can add a second infrastructure to your enterprise. This infrastructure should contain a metadata repository and, optionally, Oracle Management Server. It should be configured to use the Single Sign-On server and Oracle Internet Directory from the original infrastructure. The second Oracle Management Server should be configured to use the metadata repository from the primary infrastructure, so there is no load balancing or failover of Oracle Enterprise Manager at the repository level.

To add an infrastructure, use the Oracle Universal Installer to install the infrastructure. Choose a custom configuration that only configures the metadata repository, and, if desired, Oracle Management Server. Supply the Single Sign-On and Oracle Internet Directory information from your original infrastructure.

After the installation, the metadata repository in the new infrastructure is registered in your enterprise. During subsequent middle-tier application server installations, you should specify the Single Sign-On server and Oracle Internet Directory in your original (or primary) infrastructure. You will be given a choice of which metadata repository to use. Note that you cannot change any pre-existing middle-tier installations to use the metadata repository in the new infrastructure.

Associating an Instance with an Infrastructure (Joining a Farm)

If you have a J2EE and Web Cache instance that is not associated with an infrastructure and you would like to use Oracle9iAS Single Sign-On or clustering, you can associate the instance with an existing infrastructure using the Enterprise Manager Web site. This operation also causes the instance to join the farm.

You can associate an instance with an infrastructure using the Enterprise Manager Web site:

1. Navigate to the Instance Home Page on the Enterprise Manager Web site. Scroll to the Administration section.
2. Select **Use Infrastructure**. This launches the Use Infrastructure wizard.
3. The wizard guides you through the steps for using an infrastructure and requests the following information:
 - Oracle9iAS Single Sign-On host and port number. The port number is the same as the HTTP Server port number (default 7777).
 - Oracle Internet Directory host and port number (default port number is 389).
 - Metadata Repository host, listener port number (always 1521), and service name (default iasdb).

See Also: ["How Application Server Installations Use An Infrastructure"](#) on page 2-9.

Changing the Schema Used by a Component

You can change the database schema used by an individual component. This operation changes the schema a component points to, but does not create the schema or load any data into it. It is primarily used for migration purposes. For example, you may want to point a component to a schema used by a pre-9.0.2 Oracle application server that has been upgraded to work with Oracle9iAS version 9.0.2.

To change the database used by a component with Enterprise Manager Web site:

1. Navigate to the Instance Home Page. Scroll to the Administration section.
2. Select **Configure Schema**. This opens the Configure Schema Page.
3. Click the radio button in the Select column for the schema.
4. Click **Change Schema**. This opens the Change Schema Page.
5. Enter the new schema name, password, and database connect string.
6. Click **OK**.

See Also: *Migrating from Oracle9iAS Release 1 (1.0.2.2.x) to Release 2 (9.0.2)*

Changing Component Schema Passwords

Some application server components use schemas in the metadata repository or customer databases. The schema passwords are generated and maintained automatically by the application server, and are not usually known by administrators.

You can reset a schema password if you need direct access to the schema. You must reset most component schema passwords using Enterprise Manager Web site, because the change must be registered with the component and the database.

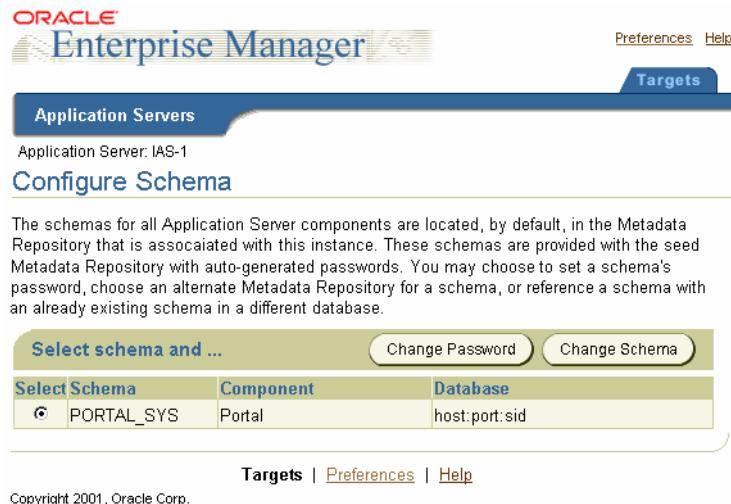
See Also: [Appendix C, "Metadata Repository Schemas"](#) for a table showing which schemas must be changed using Enterprise Manager Web site, and which must be change directory in the database

To change a schema password using Enterprise Manager Web site:

1. Navigate to the Instance Home Page. Scroll to the Administration section.

2. Select **Configure Schema** to open the Configure Schema page ([Figure 10–1](#)).

Figure 10–1 Configure Schema Page



3. Click the button in the Select column for the component.
4. Click **Change Password**. This opens the Change Password page.
5. Type a new password in the **Password** and **Confirm Password** fields.
6. Click **OK**.
7. If the schema is in the metadata repository, you do not need to perform this step and are finished. If the schema is in a customer database, you must manually change the password in the database. This is because the Enterprise Manager Web site can only update the metadata repository.

To change the password using SQL*Plus, login to the customer database as a user with SYSDBA privileges and run the following command:

```
sql> alter user schema identified by new_password
```

Note: If you change the ORASSO schema password, you must restart Oracle HTTP Server.

11

Managing the Infrastructure

This chapter provides an introduction to managing an infrastructure, and describes common management tasks for each piece of an infrastructure.

It contains the following topics:

- [Basic Infrastructure Administration](#)
- [Managing the Infrastructure Instance](#)
- [Managing Oracle9iAS Single Sign-On](#)
- [Managing the Metadata Repository](#)
- [Managing Oracle Internet Directory](#)

Basic Infrastructure Administration

The Oracle9iAS Infrastructure is a type of application server installation that provides centralized security, management services, and data repositories for middle-tier application server installations. Typically, there is one infrastructure in an application server enterprise, and all middle-tier application servers are configured to use it.

See Also: "The Oracle9iAS Infrastructure" on page 2-7 for more information about infrastructures

An infrastructure is custom-tuned and configured to support middle-tier application server installations. Many of the services provided by an infrastructure are configured automatically during installation and used programmatically by middle-tier application servers.

It is possible to run an infrastructure with very little manual intervention; however, you should perform the following basic management tasks on every infrastructure:

- [Starting and Stopping an Infrastructure](#)
- [Backing up and Restoring an Infrastructure](#)

The subsequent sections in this chapter provide an introduction to common management tasks for each piece of an infrastructure.

Starting and Stopping an Infrastructure

Starting and stopping an infrastructure involves starting and stopping the pieces of an infrastructure in the proper order. You must perform an orderly shutdown and startup of an infrastructure every time you reboot a host on which an infrastructure resides.

You should not stop an infrastructure unless all application server instances using the infrastructure are stopped. Similarly, you must start an infrastructure before starting any application server instances that use it.

Whenever you reboot a system that hosts an infrastructure, follow these steps:

1. Stop all application server instances that use the infrastructure.
2. Stop the infrastructure.

See Also: "[Stopping an Infrastructure](#)" on page 5-26 for the steps for performing an orderly shutdown of an infrastructure

3. Reboot the host.
4. Start the infrastructure.

See Also: ["Starting an Infrastructure"](#) on page 5-15 for detailed steps for performing an orderly startup of an infrastructure

5. Start all application server instances that use the infrastructure.

Backing up and Restoring an Infrastructure

To ensure that you can make a full recovery from media failures, you should perform regular backups of an infrastructure. To back up an infrastructure, you must back up each of the following:

- The Oracle home in which the infrastructure is installed
- Oracle Internet Directory
- Metadata Repository

See Also: [Appendix H, "Backup and Restore"](#) for more information

Managing the Infrastructure Instance

Every infrastructure contains a J2EE and Web Cache instance, also referred to as the *infrastructure instance*. The infrastructure instance is automatically configured and tuned to support infrastructure services; Oracle HTTP Server provides a Web server for the infrastructure, and OC4J provides a J2EE platform.

Warning: The infrastructure instance is intended only to support infrastructure services. Oracle does not support using the infrastructure instance for customer application deployment.

You can manage an infrastructure instance using its Instance Home Page on the Enterprise Manager Web site, just as you do any other application server instance.

To access the Instance Home Page for an infrastructure instance:

1. Navigate to the Enterprise Manager Web site on the infrastructure host. This opens the Farm Home Page for the metadata repository in the infrastructure on that host. This is because an infrastructure instance is automatically added to the farm for its metadata repository.
2. On the Farm Home Page, scroll to the Standalone Instances section and select the link for the infrastructure instance in the Name column. This opens the Instance Home Page for the infrastructure instance.

Using the Instance Home Page, you can perform standard instance management tasks, such as starting and stopping the instance, monitoring the instance and its components, and changing the `ias_admin` password. The Instance Home Page also provides configuration and deployment capabilities, however, you should not change the configuration settings in an infrastructure instance because it has been pre-configured to support the infrastructure, and you should not deploy applications in an infrastructure. Also, you should not add an infrastructure instance to a cluster.

Managing Oracle9iAS Single Sign-On

Oracle9iAS Single Sign-On enables users to login to application server features and access applications using a single user name and password. The benefits of using Single Sign-On include:

- **Reduced administrative costs:** The Single Sign-On server eliminates the need to support multiple accounts and passwords.
- **Convenient login:** Users do not have to maintain a separate user name and password for each application that they access.
- **Increased security:** When a password is required only once, users are less likely to use simple, easily exposed passwords, or to write these passwords down.

About Single Sign-On Installation and Configuration

When you install the primary infrastructure, you must configure Oracle9iAS Single Sign-On. The Single Sign-On service in your primary infrastructure should be the only Single Sign-On service in your enterprise. If you install any additional infrastructures in your enterprise, you should not configure Single Sign-On.

During the primary infrastructure installation, Single Sign-On is automatically configured to use schemas in the metadata repository and is fully integrated with Oracle Internet Directory. It is ready for use when the installation is complete.

About Single Sign-On Administrators

Immediately after an infrastructure installation only one Single Sign-On administrator exists: `orcladmin`. The `orcladmin` user is the super-user for Oracle Internet Directory and the `orcladmin` password is requested and set during the infrastructure installation.

After installation, you can grant Single Sign-On administrator privileges to any user by adding the user to the `IASAdmins` group in Oracle Internet Directory. The `orcladmin` user and all existing members of the `IASAdmins` group can add users to the `IASAdmins` group.

See Also: ["Adding a User to a Group" on page 12-12](#)

Single Sign-On administrators have full privileges for Single Sign-On server. Using the Single Sign-On administration pages, you can do the following:

- Configure Single Sign-On server settings.
- Administer partner applications, which are Web applications that delegate authentication to the Single Sign-On server.
- Administer external applications, which are Web applications that perform their own authentication using HTML forms.

See Also: [Oracle9iAS Single Sign-On Administrator's Guide](#) for complete information on managing the Single Sign-On server

Accessing the Single Sign-On Server

You can access the Single Sign-On Administration Home Page using a URL of the following form:

`http://host:port/pls/Single_Sign-On_DAD`

where

- *host* is the computer on which the Single Sign-On server is located
- *port* is the HTTP Server port number for the infrastructure instance to which the Single Sign-On server belongs
- *Single_Sign-On_DAD* is the database access descriptor for the Single Sign-On schema (default is `orasso`)

For example:

`http://host:7777/pls/orasso`

You can log in to Single Sign-On from the Administration Home Page by clicking **Login** in the upper right corner, and supplying a Single Sign-On administrator username and password.

Configuring Single Sign-On for Multilingual Support

Oracle9iAS Single Sign-On supports 29 languages. Only one, English, is installed by default. To install additional languages, execute the following command:

```
(UNIX) ORACLE_HOME/jdk/bin/java -jar ORACLE_HOME/sso/lib/ossoca.jar langinst  
lang make_lang_avail ORACLE_HOME  
(Windows) ORACLE_HOME\jdk\bin\java -jar ORACLE_HOME\sso\lib\ossoca.jar langinst  
lang make_lang_avail ORACLE_HOME
```

For the variable *lang*, substitute the code for the language to be installed. For the variable *make_lang_avail*, substitute 1 if you want to make the language available. Substitute 0 if you want to make the language unavailable.

See Also: Table 3-2 "Oracle9iAS Portal Languages" in *Oracle9i Application Server Globalization Support Guide* for a list of language codes, and "Configuring National Language Support" in Chapter 2 of *Oracle9iAS Single Sign-On Administrator's Guide* for more information

Managing the Metadata Repository

The metadata repository is an Oracle9*i* Enterprise Edition database that is pre-seeded with schemas used by many middle-tier application server components, Oracle Internet Directory, and Oracle Management Server. The metadata repository provides a single, ready-to-use database that greatly simplifies component configuration. You can designate a metadata repository during a middle-tier application server installation, and components in that installation will automatically use schemas in that metadata repository. Many Oracle9*i*AS demos also use metadata repository schemas.

Warning: Use of the metadata repository requires a database license. An Oracle9*i* Enterprise Edition license is required to support the following:

- Use of the metadata repository to run Oracle9*i*AS Wireless
- Use of the metadata repository to run Oracle9*i*AS Clickstream Intelligence
- Use of the replication feature of Oracle Internet Directory

All other uses of the metadata repository for Oracle9*i*AS require an Oracle9*i* Standard Edition license.

About Metadata Repository Installation and Configuration

A metadata repository is installed and configured with every infrastructure installation and uses the same Oracle home as the infrastructure instance. It contains schemas to support Oracle9*i*AS components.

See Also: [Appendix C, "Metadata Repository Schemas"](#) for a list of schemas

The metadata repository is always configured to use port 1521 and has a service name of the form `iasdb.host.domain` (for the first infrastructure on a host) and `iasdbo.host.domain` for any subsequent infrastructures on that host). You should never change these values. If you have another database on the host using the same service name, you must change it before installing the metadata repository.

If you have another application on the host using port 1521, you must do one of the following *before* installing the infrastructure:

- If the other application is an Oracle Net listener using an Oracle9*i* database, no action is required. The metadata repository will use the existing Oracle Net listener.
- If the other application is an Oracle Net8 listener using an Oracle8*i* database, you must change the Net8 listener to use a different port than 1521, or you must upgrade it to the Oracle9*i* Net listener that runs with the metadata repository.

To upgrade the Net8 listener, stop the Net8 listener before installing the infrastructure. After the infrastructure installation is complete, migrate the Oracle Net8 listener configuration specific to your Oracle8*i* database into the configuration for the metadata repository Oracle Net listener, and restart the Oracle Net listener. The Oracle Net listener will support the Oracle8*i* database as well as the metadata repository.

See Also: *Oracle9*i* Application Server Installation Guide* for instructions on upgrading an existing Net8 listener

- If the other application is not an Oracle Net or Oracle Net8 listener, reconfigure it to use a different port.

Starting and Stopping the Metadata Repository

You can start and stop the metadata repository using SQL*Plus or Oracle Enterprise Manager. You should always start the metadata repository before starting any other pieces of the infrastructure, and stop it only after all other pieces of the infrastructure have been stopped.

See Also: ["Starting an Infrastructure" on page 5-4](#) for information on starting and stopping the metadata repository

Changing Metadata Repository Passwords

This section describes how to change passwords for two types of schemas in the metadata repository:

- [Schemas Accessed Directly by Administrators](#)
- [Schemas Accessed Internally by Components](#)

Schemas Accessed Directly by Administrators

You can change the passwords for schemas that are accessed directly by administrators at any time, using your preferred method. The following schemas fall into this category:

- **SYS** (initial password is `change_on_install`)
- **SYSTEM** (initial password is `manager`)
- **SCOTT** (initial password is `tiger`)

This schema is used by demos and it is recommended that you do not change the password or some demos may not work.

- **OEM_REPOSITORY** (initial password is `oem_repository`)

This schema is used by Oracle Management Server. If you change the password in the database, you must update it in the Oracle Management Server configuration by running the Enterprise Manager Configuration Assistant and choosing the option to "Use an Existing Repository." By selecting this option and entering the new password, the Oracle Management Server configuration is updated.

Schemas Accessed Internally by Components

Schemas that are accessed internally by components are assigned a randomly generated password at install time that is known only by the components. If you would like to reset any of these passwords, you must do it using the Enterprise Manager Web site. This is so the change is registered with the component as well as in the metadata repository.

See Also: [Appendix C, "Metadata Repository Schemas"](#) for a list of schemas used by components and information on which method to use to change their passwords and ["Changing Component Schema Passwords"](#) on page 10-5 for instructions on changing passwords using the Enterprise Manager Web site

Relocating Metadata Repository Datafiles to a Different Directory

By default, the metadata repository is installed in the same Oracle home as the infrastructure. If you do not have enough space in the infrastructure Oracle home, you may want to relocate some or all of the metadata repository datafiles to a different directory.

You can follow the steps in this section to relocate the datafiles for one or more tablespaces. As an example, the steps show how to relocate the `users` and `portal` tablespaces from an original infrastructure Oracle home named `/infra_home` to a new directory named `/new_directory`.

See Also: *Oracle9i Database Administrator's Guide* for more information about database concepts and relocating datafiles

Assume the following:

- The metadata repository has tablespaces named `users` and `portal` that are made up of datafiles all located in the same directory.
- The datafiles of the `users` and `portal` tablespaces are to be relocated to a different directory on a different filesystem or disk.
- You are currently connected to the metadata repository with administrator privileges. You must have the `ALTER DATABASE` system privilege to relocate datafiles.
- You have a current backup of the metadata repository.

Complete the following steps:

1. Identify the datafile names of interest.

The following query of the data dictionary view `DBA_DATA_FILES` lists the datafile names and respective sizes (in bytes) of the `users` and `portal` tablespaces:

```
SELECT FILE_NAME, BYTES FROM DBA_DATA_FILES  
WHERE TABLESPACE_NAME = 'USERS' OR TABLESPACE_NAME = 'PORTAL';
```

FILE_NAME	BYTES
<code>/infra_home/oradata/iasdb/portal.dbf</code>	78643200
<code>/infra_home/oradata/iasdb/users01.dbf</code>	96993280

2. Stop the entire infrastructure, including the metadata repository. Before stopping the infrastructure, you must stop all middle-tier application server instances that use it.

See Also: ["Stopping an Infrastructure"](#) on page 5-26

3. Start a metadata repository instance and mount the database without opening it. To do this, connect to the metadata repository as a user with SYSDBA privileges and run this command:

```
STARTUP MOUNT;
```

4. Move the datafiles to their new location using the operating system. For example:

(UNIX)

```
mv /infra_home/oradata/iasdb/portal.dbf /new_directory/portal.dbf  
mv /infra_home/oradata/iasdb/users01.dbf /new_directory/users01.dbf
```

(Windows)

```
rename C:\infra_home\oradata\iasdb\portal.dbf D:\new_directory\portal.dbf  
rename C:\infra_home\oradata\iasdb\users01.dbf D:\new_directory\users01.dbf
```

Note: You can execute an operating system command to copy a file by using the SQL*Plus HOST command.

5. Use ALTER DATABASE to rename the file pointers in the database's control file.

```
ALTER DATABASE  
  RENAME FILE  '/infra_home/oradata/iasdb/portal.dbf',  
               '/infra_home/oradata/iasdb/users01.dbf'  
  TO           '/new_directory/portal.dbf',  
               '/new_directory/users01.dbf';
```

The new files must already exist; this statement does not create the files. Also, always provide complete filenames (including their paths) to properly identify the old and new datafiles. In particular, specify the old datafile name exactly as it appears in the DBA_DATA_FILES view of the data dictionary.

6. Back up the metadata repository. After making any structural changes to a database, always perform an immediate and complete backup.
7. Shutdown the metadata repository.
8. Start the entire infrastructure.

See Also: "Starting an Infrastructure" on page 5-15

Managing Oracle Internet Directory

Oracle Internet Directory is an online directory, which is a specialized database that stores information in a hierarchical format for fast lookup and retrieval. It implements Lightweight Directory Access Protocol (LDAP), Version 3, an industry-standard protocol for accessing online directory services.

Most of the management tasks associated with Oracle Internet Directory are related to application server security. These tasks include:

- Managing User and Group Privileges
- Delegating Administration Tasks
- Managing Password Policies

See Also: Concepts and tasks for managing application server security using Oracle Internet Directory are discussed in [Chapter 12, "Managing Oracle Internet Directory"](#)

Starting and Stopping Oracle Internet Directory

You should always start and stop Oracle Internet Directory as part of starting and stopping the infrastructure instance using the Enterprise Manager Web site. This ensures that the directory is started using the proper options and in the correct order with the rest of the infrastructure instance components.

Configuring Oracle Internet Directory and Single Sign-On on Separate Hosts

In most cases, all the pieces of an infrastructure reside on a single host in one Oracle home. You can, however, configure Oracle9iAS Single Sign-On and Oracle Internet Directory on separate hosts. You can do this at installation time using the following steps:

1. Install an infrastructure on Host 1 and choose a custom configuration. Select Oracle Internet Directory and the metadata repository to be configured (deselect Oracle9iAS Single Sign-On).
2. Install an infrastructure on Host 2 and choose a custom configuration. Select Oracle9iAS Single Sign-On and the metadata repository to be configured (deselect Oracle Internet Directory).
3. Point the Oracle9iAS Single Sign-On configuration on Host 2 to the Oracle Internet Directory configuration on Host 1.

If you plan to install copies of Oracle Internet Directory on different hosts to form a directory replication network, configure Oracle9iAS Single Sign-On with only one of the Oracle Internet Directory copies. The remaining copies of Oracle Internet Directory must not associate with any Oracle9iAS Single Sign-On configuration.

See Also: *Oracle Internet Directory Administrator's Guide* for more information on directory replication

Managing Oracle Internet Directory

This chapter describes how to manage user privileges in Oracle Internet Directory. It contains the following topics:

- [Oracle Internet Directory Security](#)
- [Delegating Administration Tasks](#)
- [Administering Users](#)
- [Administering Groups](#)
- [Administering Privileges of Oracle9iAS Components](#)
- [Installing and Configuring Oracle9iAS](#)
- [Upgrading Oracle Internet Directory for Oracle9iAS](#)
- [Password Policies](#)
- [Modifying Oracle Internet Directory Passwords](#)
- [Changing Instance Passwords in Oracle Internet Directory](#)
- [Removing an Application Server Instance From Oracle Internet Directory](#)

Oracle Internet Directory Security

This chapter describes administering Oracle Internet Directory with respect to Oracle9iAS. If you need details on specific Oracle Internet Directory features and tools, or if you need background information on Oracle Internet Directory, see the *Oracle Internet Directory Administrator's Guide*.

Oracle Internet Directory is an online directory, which is a specialized database that stores information in a hierarchical format for fast lookup and retrieval. It implements Lightweight Directory Access Protocol (LDAP) version 3, an industry-standard protocol for accessing online directory services.

Oracle9iAS can use an Oracle Internet Directory that is already in use in your enterprise. In such cases, you typically work with the Oracle Internet Directory administrator to set up and manage the Oracle9iAS nodes. The Oracle Internet Directory administrator should read this chapter to be aware of how Oracle9iAS uses Oracle Internet Directory.

About the Terms "Subscriber" and "Enterprise"

In Oracle Internet Directory, "subscriber" has the same meaning as "enterprise":

- If you set up Oracle Internet Directory to host other enterprises, each enterprise is a subscriber. For example, if you (company A) host companies X and Y on your Oracle Internet Directory instance, then X and Y are subscribers in your Oracle Internet Directory. Each subscriber (that is, enterprise) can have its own subscriber administrator.
- If you are using Oracle Internet Directory to contain data from your enterprise only, your Oracle Internet Directory has only one subscriber, namely, your enterprise.

This chapter uses "subscriber" and "enterprise" interchangeably.

How Oracle9iAS Uses Oracle Internet Directory

Oracle9iAS uses containers, groups, and privileges in Oracle Internet Directory. You can use them to administer users and groups. You can delegate privileges to other users ("product administrators") so that they can manage their product. For example, you can give privileges to a user to be a Portal administrator. You can have one or more administrators per product.

When you install Oracle9iAS Infrastructure, the installer configures the metadata repository database to use Oracle Internet Directory. Oracle9iAS creates its own area in Oracle Internet Directory. This enables all Oracle9iAS components (for example, Portal, Wireless, and Delegated Administration Service) to access the same data. It also enables an enterprise to centralize security by managing users and applications under one server.

This model is different from Oracle9iAS Release 1.x. In Release 1.x, each component had its own repository of users.

New Oracle Internet Directory Instance or Existing Oracle Internet Directory Instance

The Oracle Internet Directory that Oracle9iAS uses can be a newly created instance, or you can direct the installer to use an existing Oracle Internet Directory. For example, if you have an existing Oracle Internet Directory instance that you use as a corporate directory and that contains entries for users, and you plan for those users to access applications from Oracle9iAS, then it might make sense to share that Oracle Internet Directory instance with Oracle9iAS.

To distinguish the two types of configuration, this chapter uses the following terms:

- **Newly created Oracle Internet Directory:** The installer installs a new instance of Oracle Internet Directory. Typically, this Oracle Internet Directory instance is not shared with third-party software. The installer configures the Oracle Internet Directory instance with areas, groups, and privileges required by Oracle9iAS. You need to create user entries for this new Oracle Internet Directory.

In a newly created Oracle Internet Directory, you (the Oracle9iAS administrator) perform Oracle Internet Directory administrative duties.

- **Existing Oracle Internet Directory:** You point the installer to use an existing Oracle Internet Directory instance. The installer creates an area for Oracle9iAS in the Oracle Internet Directory instance, but does not create privileges. You and the Oracle Internet Directory administrator need to assign privileges to users and Oracle9iAS components based on the security policy of your enterprise.

In an existing Oracle Internet Directory, you and the Oracle Internet Directory administrator work together to manage the Oracle Internet Directory instance.

Note: If you are using an existing Oracle Internet Directory instance, you need to upgrade it to Release 9.0.2 before Oracle9iAS can use it. See *Oracle Internet Directory Administrator's Guide* for instructions on how to perform the upgrade.

See also "[Upgrading Oracle Internet Directory for Oracle9iAS](#)" on page 12-17 for upgrade details specific to using Oracle Internet Directory with Oracle9iAS.

Superuser and Administrators in Oracle Internet Directory

Oracle Internet Directory contains the following superuser and administrators:

- Oracle Internet Directory superuser (DN: cn=orcladmin)

This superuser is allowed to perform all operations in Oracle Internet Directory.

Note that you cannot log in as this superuser using applications that use Oracle9iAS Single Sign-On (such as the Delegated Administration Service application). If you enter "orcladmin" as the user in such applications, Oracle9iAS Single Sign-On authenticates you as the subscriber administrator, whose DN is "cn=orcladmin, cn=user, <Enterprise DN>". To log in as the orcladmin superuser, you use Oracle Directory Manager.

For information on Oracle Directory Manager, see the *Oracle Internet Directory Administrator's Guide*.

- Subscriber administrator (DN: cn=orcladmin, cn=users, <Enterprise DN>)

This administrator is allowed to perform all operations in the enterprise subtree. For example, this user can create new users using Delegated Administration Service and assign them privileges.

You can have more than one subscriber administrator. This is typical in cases where you are hosting other enterprises.

For information on Delegated Administration Service, see the *Oracle Internet Directory Administrator's Guide*.

During Oracle9iAS installation, you specify the password for Oracle Internet Directory superuser and subscriber administrators. The same password is used for these users.

Relationship Between Oracle Internet Directory and the `ias_admin` User

When you manage Oracle9iAS using the Oracle Enterprise Manager Web site, you log in as the `ias_admin` user. This user is not stored in Oracle Internet Directory. Instead, information about this user (such as its password) is stored separately for each host. The `ias_admin` password is the same for all application server installations on one host.

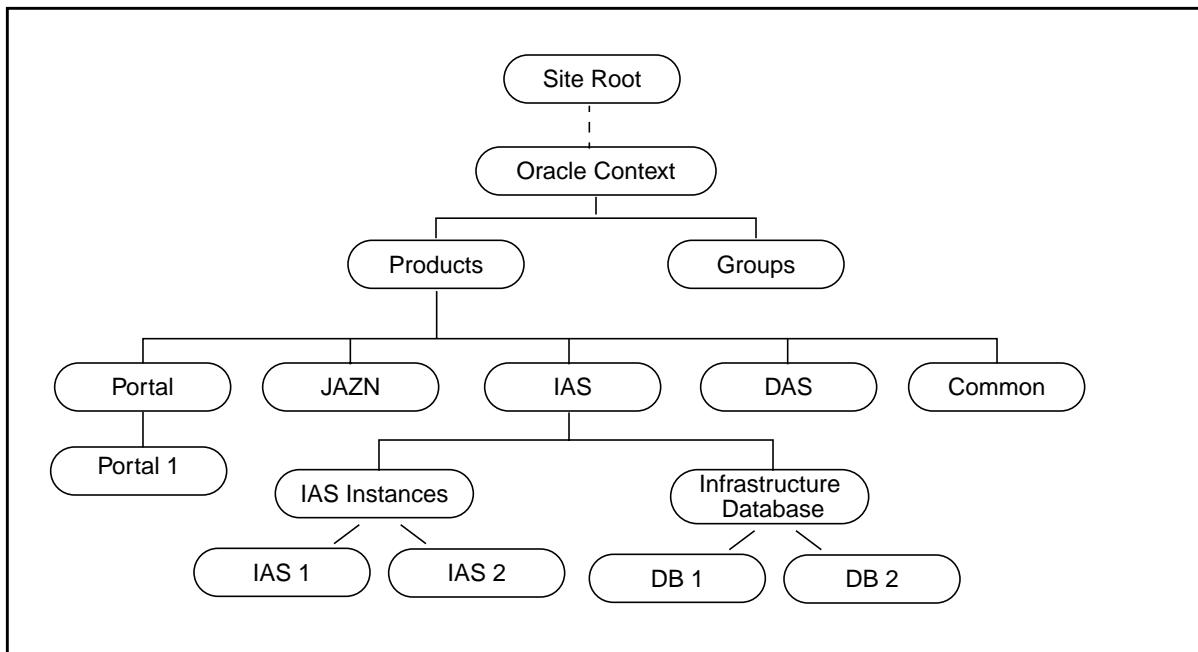
See Also: ["Changing the `ias_admin` Password" on page 3-3](#)

To manage users, groups, and privileges in Oracle Internet Directory, you do not need to know the `ias_admin` user password because you do not use Oracle Enterprise Manager to perform these management tasks. However, the `ias_admin` password may be used internally to set component passwords within Oracle Internet Directory. For this reason, it is important to adhere to the Oracle Internet Directory password policy when setting the `ias_admin` password.

See Also: ["Password Policies" on page 12-18](#) and ["Changing the `ias_admin` Password" on page 3-12](#)

Oracle9iAS Tree in Oracle Internet Directory

When you install Oracle9iAS Infrastructure, the installer creates the following tree in Oracle Internet Directory ([Figure 12-1](#)):

Figure 12–1 Oracle9iAS Tree in Oracle Internet Directory

Oracle9iAS stores its information in Oracle Context, which is also used by other Oracle products. Immediately below Oracle Context are Products and Groups.

Under Products, there is an area for each component. Each component area has its own component-specific metadata. You do not have to edit anything under Products.

Under Groups, there is a node for each access control group. Oracle9iAS uses groups to assign privileges. To enable a user to have a certain privilege, you add the user to the appropriate group. For example, to enable a user to create new users, you add the user to the oracleDASCreateUser group.

You can assign users to have certain privileges at each level of the tree. For example, you can have Oracle Context administrators, IAS administrators, and Portal administrators.

Tasks

The main administration tasks that you have to perform depend on which mode you are running in. In an existing Oracle Internet Directory, the Oracle Internet Directory administrator performs some of the tasks. In a newly created Oracle

Internet Directory, the Oracle Internet Directory installation creates a default configuration to help you perform the tasks.

The following table shows who should perform each task:

Table 12–1 Tasks

Task	Newly created Oracle Internet Directory	Existing Oracle Internet Directory
Upgrade Oracle Internet Directory to Release 9.0.2 See " Upgrading Oracle Internet Directory for Oracle9iAS ".	n/a	Oracle Internet Directory administrator
Add users to Oracle Internet Directory See " Creating a User in Oracle Internet Directory ".	You do this	Oracle Internet Directory administrator
Add users to groups See " Adding a User to a Group ".	You do this	Oracle Internet Directory administrator
Associate groups with privileges See the <i>Oracle Internet Directory Administrator's Guide</i> .	Done for you	Oracle Internet Directory administrator
Create privileges See the <i>Oracle Internet Directory Administrator's Guide</i> .	Done for you	Oracle Internet Directory administrator
Make some users administrators See " Adding a User to a Group ".	You do this	You do this
Set the password policy See " Modifying the Password Policy ".	You do this	Oracle Internet Directory administrator
Change the orcladmin (superuser) password See " Modifying the orcladmin Password ".	You do this	Oracle Internet Directory administrator
Change the application server instance password See " Changing Instance Passwords in Oracle Internet Directory ".	You do this	Oracle Internet Directory administrator

Tools for Managing Oracle Internet Directory

You can perform the tasks described in this chapter using Oracle Directory Manager or Delegated Administration Service. These are tools that you can use to manage Oracle Internet Directory. Oracle Directory Manager is a Java-based tool, and Delegated Administration Service is a browser-based tool.

For a complete description of all the tools that you can use with Oracle Internet Directory, see the *Oracle Internet Directory Administrator's Guide*.

Oracle Directory Manager

Oracle Directory Manager is a Java-based tool for administering Oracle Internet Directory. You start Oracle Directory Manager with the following command:

(UNIX) `ORACLE_HOME/bin/oidadmin`

(Windows) From the Start menu, select Programs -> `ORACLE_HOME` -> Integrated Management Tools -> Oracle Directory Manager

The first time you start Oracle Directory Manager, an alert prompts you to connect to a directory server. To connect to a directory server:

1. Click **OK**. The Directory Server Connection dialog box appears.
2. Enter the name of the host on which Oracle Internet Directory resides, and the Oracle Internet Directory port number (the default is 389) in the Directory Server Connection dialog box. Click **OK**.
3. Enter the username and password.
If you are logging in for the first time, you need to log in as the superuser (`cn=orcladmin`).
4. Click **Login**. Oracle Directory Manager appears.

See Also: "Using Oracle Directory Manager" in Chapter 4 of *Oracle Internet Directory Administrator's Guide* for more information on starting and navigating Oracle Directory Manager.

Delegated Administration Service

Delegated Administration Service is a Web-based tool that enables you to create and manage users and groups. Users can use this tool to update information about themselves (such as changing their passwords).

To access Delegated Administration Service, point your browser to:

`http://host:port/oiddas`

host specifies the machine running Oracle HTTP Server.

port specifies the port that Oracle HTTP Server is listening on.

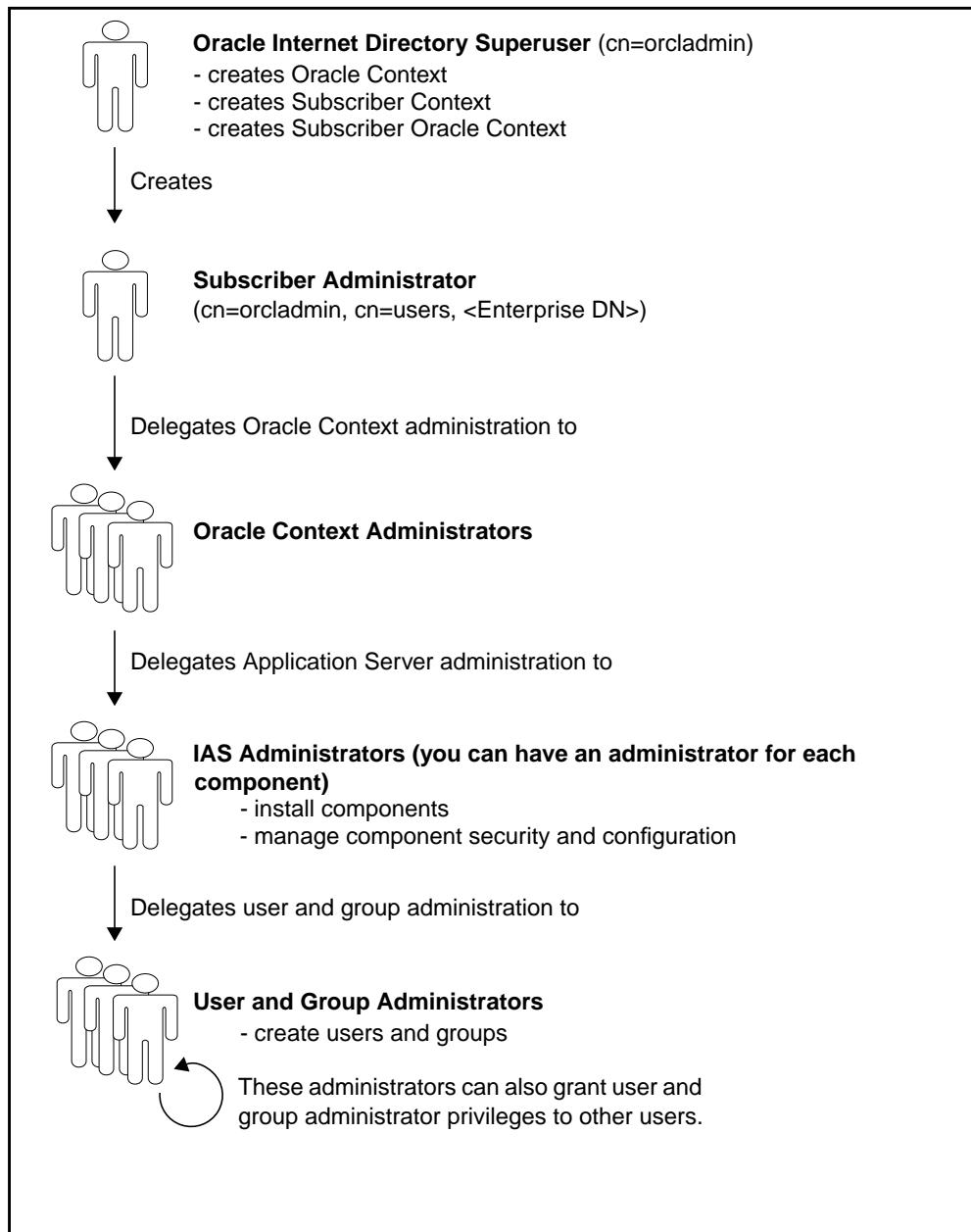
Delegating Administration Tasks

During installation of a new Oracle Internet Directory instance, the Oracle Internet Directory superuser creates the Oracle Context, the Subscriber (or Enterprise) Context, and the subscriber administrator with the appropriate privileges.

You can use the subscriber administrator to delegate administration of the Oracle Context tree to Oracle Context administrators. They, in turn, can delegate administration of Oracle9iAS and its components to Oracle9iAS administrators. The Oracle9iAS administrators have the necessary privileges to install all Oracle9iAS components. The Oracle9iAS administrators can delegate administration of users and groups to other users. [Figure 12–2](#) shows the delegation flow.

If you are working in an existing Oracle Internet Directory, you need to work with the Oracle Internet Directory administrator to ensure that you have the following privileges:

- Privilege to administer the IAS area by being a member of the IASAdmins group: This enables you to install and configure Oracle9iAS components.
- Privilege to delegate privileges to other users: This enables you to delegate privileges to application administrators (for example, Portal administrator).

Figure 12–2 Delegation flow

Administering Users

All users who access Oracle9iAS should have a user entry in Oracle Internet Directory. This requirement is mandatory for users who access applications that check authentication and authorization.

Newly Created Oracle Internet Directory

Using the subscriber administrator account (which was created during installation) and Delegated Administration Service, you can create users and add them to groups, thus delegating privileges to those users.

You can also delegate user-related operations to other users. To do this, add the users to the appropriate groups, as shown in the following table:

Table 12–2 User operations

To allow a user to do this operation:	Add the user to this group:
Create users	User Create (DN: cn=oracleDASCreateUser, cn=groups, <Oracle Context DN>)
Edit user properties	User Edit (DN: cn=oracleDASEditUser, cn=groups, <Oracle Context DN>)
Delete users	User Delete (DN: cn=oracleDASDeleteUser, cn=groups, <Oracle Context DN>)

Existing Oracle Internet Directory

If the Oracle Internet Directory instance also serves as a corporate directory, then it already has users in it. The Oracle Internet Directory administrator is responsible for managing the users.

Creating a User in Oracle Internet Directory

To create users using Delegated Administration Service:

1. Log in to Delegated Administration Service. You can log in as orcladmin (cn=orcladmin), or as a user who is a member of the IASAdmins or the oracleDASCreateUser group.

In a browser, go to:

`http://host:port/oiddas`

host specifies the machine running Oracle HTTP Server.

port specifies the port that Oracle HTTP Server is listening on.

2. Select the Directory tab.
3. Log in to Oracle Internet Directory.
4. Select **Users** in the upper left side of the page.
5. Click **Create**. This displays the Create User page.
6. Enter information for the user in the fields.
7. Click **Submit**.

For details on Delegated Administration Service, see Chapter 9, "The Delegated Administration Service", in the *Oracle Internet Directory Administrator's Guide*.

You can also create users in Oracle Internet Directory using Oracle Directory Manager. For details, see the section "Adding Entries by Using Oracle Directory Manager" in Chapter 7, "Managing Directory Entries", of the *Oracle Internet Directory Administrator's Guide*.

If you already have users in Oracle Internet Directory, you can create new users using the procedure described in the section "Adding an Entry by Copying an Existing Entry in Oracle Directory Manager" in Chapter 7 of *Oracle Internet Directory Administrator's Guide*.

Adding a User to a Group

To add a user to a group using Delegated Administration Service:

1. Log in to Delegated Administration Service. You can log in as orcladmin (*cn=orcladmin*), or as a user who is a member of the IASAdmins or the oracleDASUserPriv group. You can also log in as the owner of the group.

In a browser, go to:

`http://host:port/oiddas`

host specifies the machine running Oracle HTTP Server.

port specifies the port that Oracle HTTP Server is listening on.

2. Select the Directory tab.

3. Click **Groups** in the upper left side of the page.
4. Search for the group to which you want to add users.
5. Select the group and click **Edit**. This displays the Edit Group page.
6. Click **Add User Member** in the Members section. This displays the Search and Select window.
7. Search for the user that you want to add to the group.
8. Click **Select**. This closes the Search and Select window.
9. Click **Submit** on the Edit Group page.

You can also use Oracle Directory Manager to add users to groups. See Chapter 7 of the *Oracle Internet Directory Administrator's Guide* for details.

Administering Groups

Group memberships define the privileges a user has. To give privileges to a user, add the user to the appropriate groups.

Newly Created Oracle Internet Directory

Operations that you can perform on groups are creating groups, deleting groups, and editing properties of groups. To allow users to perform these operations, add the users to the appropriate groups, as shown in the following table:

Table 12–3 Group operations

To allow a user to do this operation:	Add the user to this group:
Create groups	Group Create (DN: cn=oracleDASCreateGroup, cn=groups, <Oracle Context DN>)
Edit group properties	Group Edit (DN: cn=oracleDASEditGroup, cn=groups, <Oracle Context DN>)
Delete groups	Group Delete (DN: cn=oracleDASDeleteGroup, cn=groups, <Oracle Context DN>)

Existing Oracle Internet Directory

In an existing Oracle Internet Directory, you may not have group administration privileges because the Oracle Internet Directory administrator has already set up the necessary groups.

Group Details

In a newly created Oracle Internet Directory, the installer creates groups and associates them with privileges.

In an existing Oracle Internet Directory, the Oracle Internet Directory upgrade process to Release 9.0.2 creates groups mentioned in this chapter, but it does not set up any ACLs in the user namespace because this would override existing ACLs. You and the Oracle Internet Directory administrator can grant the necessary privileges to the groups, or you can associate privileges to existing groups. This section describes the privileges for each group.

Table 12–4 Groups for component administration

Group	DN	Required Privilege
IAS Admins	cn=IASAdmins, cn=groups, <Oracle Context DN>	Complete access to the IAS node under Products in Oracle Context Add access to create Application entity objects under individual products Proxy access to these application entities
Read and modify user passwords	cn=oracleUserSecurityAdmins, cn=groups, <Oracle Context DN>	Read, write, compare, and search access to the following attributes: <ul style="list-style-type: none">▪ userpkcs12▪ orclpkcs12hint▪ userpassword▪ orclpassword▪ orclpasswordVerifier
Compare user passwords	cn=authenticationServices, cn=groups, <Oracle Context DN>	Compare access to the userpassword attribute of users
Proxy privilege	cn=userProxyPrivilege, cn=groups, <Oracle Context DN>	Proxy access on behalf of the end user
Oracle Context administration	cn=oracleContextAdmins, cn=groups, <Oracle Context DN>	Complete access within Oracle Context

Table 12–5 Groups for user administration

Group	DN	Required Privilege
Create user	cn=oracleDASCreateUser, cn=groups, <Oracle Context DN>	Add access to the users container in an enterprise
Edit user attribute	cn=oracleDASEditUser, cn=groups, <Oracle Context DN>	Write access to user entries
Delete user	cn=oracleDASDeleteUser, cn=groups, <Oracle Context DN>	Delete access to the users container in an enterprise
Delegate user administration	cn=oracleDASUserPriv, cn=groups, <Oracle Context DN>	Add access to the users container in an enterprise Delete access to the users container in an enterprise Write access to user entries

Table 12–6 Groups for group administration

Group	DN	Required Privilege
Create group	cn=oracleDASCreateGroup, cn=groups, <Oracle Context DN>	Add access to the groups container in an enterprise
Edit group attribute	cn=oracleDASEditGroup, cn=groups, <Oracle Context DN>	Write access to group entries
Delete group	cn=oracleDASDeleteGroup, cn=groups, <Oracle Context DN>	Delete access to the groups container in an enterprise
Delegate group administration	cn=oracleDASGroupPriv, cn=groups, <Oracle Context DN>	Add access to the groups container in an enterprise Delete access to the groups container in an enterprise Write access to group entries

To Determine a Group's Privileges

To determine which privileges a group has, you can use Oracle Directory Manager. See the *Oracle Internet Directory Administrator's Guide* for details on Oracle Directory Manager.

Administering Privileges of Oracle9iAS Components

Some components in Oracle9iAS need to authenticate users (verify that a user presented the correct password) or to impersonate their clients. The components need the following privileges to perform these operations.

- For a component to authenticate users, it needs to be in the Authentication Services group (DN: cn=authenticationServices, cn=groups, <Oracle Context DN>). Members of this group can authenticate users by comparing the presented password with the one stored in Oracle Internet Directory.
Members of this group include Oracle9iAS Single Sign-On, Unified Messaging, and email server.
- For a component to impersonate their end-users (that is, proxy on behalf of their clients), it needs to be in the User Proxy Privilege group (DN: cn=userProxyPrivilege, cn=groups, <Oracle Context DN>).
Components use proxying so that they are able to read data for that user.

When you install components in Oracle Internet Directory (regardless of whether the Oracle Internet Directory was newly created or existing), the installer adds them to the proper groups. You do not have to add components to groups manually.

Installing and Configuring Oracle9iAS

When you add or remove components from your Oracle9iAS installation, the installer needs to update the configuration information in Oracle Internet Directory. The user who runs the installer needs privileges in the IAS area in Oracle Internet Directory.

Newly Created Oracle Internet Directory

You must be a member of the IASAdmins group, which has privileges to administer the IAS tree in Oracle Internet Directory.

Existing Oracle Internet Directory

You need to have privileges equivalent to those in the IASAdmins group.

Upgrading Oracle Internet Directory for Oracle9iAS

This section describes the upgrade details for Oracle Internet Directory that are specific to Oracle9iAS. For instructions on how to upgrade Oracle Internet Directory to Release 9.0.2, see the *Oracle Internet Directory Administrator's Guide*.

When you install Oracle9iAS with a new Oracle Internet Directory, the installer installs a new Oracle Internet Directory Release 9.0.2 and configures it for Oracle9iAS. It creates all the groups and sets them up with the appropriate privileges. It also sets up ACLs in the users and the Oracle Context namespaces.

If you upgrade your Oracle Internet Directory to Release 9.0.2 and plan to use it with Oracle9iAS, the upgrade creates ACLs in the Oracle Context namespace, but not in the users namespace. This is to avoid overriding existing policies in the users namespace. It does, however, create groups mentioned in this chapter.

The following ACL policies are not created during the upgrade:

- privileges for the User Proxy Privilege group
- privileges for the Authentication Services group
- privileges for creating, editing, and deleting users
- privileges for creating, editing, and deleting groups

To follow the administration model described in this chapter, you and the Oracle Internet Directory administrator can create the ACLs described in "[Group Details](#)" on page 12-14 and associate them with the appropriate groups, or you and the Oracle Internet Directory administrator can associate equivalent existing ACLs with the appropriate groups.

The subscriber administrator is also not created during the upgrade process.

For instructions on how to create ACLs in the users namespace and how to associate groups with ACLs, see the *Oracle Internet Directory Administrator's Guide*.

Password Policies

Oracle Internet Directory enables you to define password policies that are compliant with the security policy of your enterprise.

About Password Policies

A *password policy* is a set of rules for setting and managing passwords. These rules include:

- The minimum number of characters a password must contain. On installation, this is set to five characters.
- The minimum number of numeric characters a password must contain. On installation, this is set to one numeric character.

Oracle Internet Directory enforces password policies for its users. You should be aware of the Oracle Internet Directory password policy for two reasons:

- You must comply with the password policy every time you change the Oracle Internet Directory superuser password.
- You must comply with the password policy when you set the password for the `ias_admin` user. Note that this user is not stored in Oracle Internet Directory; as such, its password is **not** enforced by Oracle Internet Directory. However, you must follow the same password policy because the `ias_admin` user password is used internally to create objects in the directory. If the `ias_admin` user password does not comply with the Oracle Internet Directory password policy, an error occurs.

Viewing the Password Policy

All users can view the password policy using Oracle Directory Manager:

1. Start Oracle Directory Manager with the following command:

`ORACLE_HOME/bin/oidadmin`

2. Log in to Oracle Directory Manager. If you are logging in as `orcladmin`, enter "`cn=orcladmin`" in the **User** field. If you are logging in as another user, enter "`cn=username,cn=Users,o=DEFAULT SUBSCRIBER,dc=COM`" in the **User** field. This launches the Oracle Directory Manager.
3. Expand the user entry in the System Objects frame by clicking the plus sign (+) next to it. The user entry is of the form: `username@host:port`.

4. Expand the Password Policy Management entry in the System Objects frame by clicking the plus sign (+) next to it.
5. Select the "cn=PwdPolicyEntry" entry.
If there is more than one "cn=PwdPolicyEntry" entry, select each one and examine the contents of the **Path to Password Policy Entry** field in the General Tab in the pane on the right side. Select the entry that has "o=DEFAULTSUBSCRIBER" as part of the path in the **Path to Password Policy Entry** field. Be sure to examine the full path by clicking in the **Path to Password Policy Entry** field and moving your cursor to the end of the path.
6. View the fields in the General tab, in particular the **Number of Numeric Characters in Password** field and the **Minimum Number of Characters of Password** field.

Modifying the Password Policy

Only the orcladmin user can modify the Oracle Internet Directory password policy.

1. Start Oracle Directory Manager with the following command:
`ORACLE_HOME/bin/oidadmin`
2. Log in to Oracle Directory Manager. Enter "cn=orcladmin" in the **User** field. This launches the Oracle Directory Manager.
3. Expand the user entry in the System Objects frame by clicking the plus sign (+) next to it. The user entry is of the form: *username@host:port*.
4. Expand the Password Policy Management entry in the System Objects frame by clicking the plus sign (+) next to it.
5. Select the "cn=PwdPolicyEntry" entry.

If there is more than one "cn=PwdPolicyEntry" entry, select each one and examine the contents of the **Path to Password Policy Entry** field in the General Tab in the pane on the right side.

Select the entry that has "o=DEFAULTSUBSCRIBER" as part of the path in the **Path to Password Policy Entry** field.

6. Modify the fields in the General tab, in particular the **Number of Numeric Characters in Password** field and the **Minimum Number of Characters of Password** field.
7. Click **Apply**.

Modifying Oracle Internet Directory Passwords

Individual users can change their own passwords using Delegated Administration Service or Oracle Directory Manager. Oracle Internet Directory superusers and administrators can also change the passwords of other users if they have the appropriate permissions.

Passwords are stored in the userPassword attribute.

See the *Oracle Internet Directory Administrator's Guide* for details.

Modifying the orcladmin Password

You can change the Oracle Internet Directory superuser password using ldapmodify or Oracle Directory Manager.

To modify the password of the orcladmin user in Oracle Internet Directory:

1. Start Oracle Directory Manager with the following command:

(UNIX) `ORACLE_HOME/bin/oidadmin`
(Windows) From the Start menu, select Programs -> `ORACLE_HOME` -> Integrated Management Tools -> Oracle Directory Manager

2. Log in to Oracle Directory Manager as the orcladmin user.
3. Select the `orcladmin` user entry in the System Objects frame. The user entry is of the form: `username@host:port`.
4. Select the **System Passwords** tab in the pane on the right side.
5. Enter the new password in the **Super User Password** field.
6. Click **Apply**.

Modifying a User Password

Users can log into Delegated Administration Service to modify their own passwords. The procedure is:

1. Log in to Delegated Administration Service as the user.

In a browser, go to:

`http://host:port/oiddas`

`host` specifies the machine running Oracle HTTP Server.

`port` specifies the port that Oracle HTTP Server is listening on.

2. Select the My Profile tab.
3. Log in to Oracle Internet Directory.
4. Click **Change My Password**.
5. Change your password to Oracle9iAS Single Sign-On:
 - a. In the Single Sign-On Password section, in the **Old Password** field, enter your current password.
 - b. In the **New Password** field, enter your new password, and confirm it in the **Confirm New Password** field.
 - c. Click **Submit**.

Changing Instance Passwords in Oracle Internet Directory

Each application server instance that uses an infrastructure has an entry in Oracle Internet Directory. The instance uses this entry to manage configuration information in Oracle Internet Directory.

Oracle9iAS generates random passwords for the instances in Oracle Internet Directory. You do not need to know what the passwords are because there are no procedures that you need to run that require the passwords.

However, if your corporate security policy requires that passwords be changed on a regular basis, you can use the `resetiASpasswd` tool to change the password.

Note: You cannot use Oracle Directory Manager, Delegated Administration Service, or `ldapmodify` to change the instance passwords; you can only use `resetiASpasswd`. The reason for this is that the password needs to be synchronized on the instance host and on Oracle Internet Directory.

To reset the password to a new randomly generated password, execute the following command in the Oracle home of the application server instance whose password you would like to change:

(UNIX) `ORACLE_HOME/bin/resetiASpasswd.sh cn=orcladmin password ORACLE_HOME`
(Windows) `ORACLE_HOME\bin\resetiASpasswd cn=orcladmin password ORACLE_HOME`

password is the `orcladmin` password.

ORACLE_HOME is the full path of the Oracle home for the application server instance. Note that this is identical to the Oracle home in which you run the command.

Removing an Application Server Instance From Oracle Internet Directory

When you deinstall an application server installation using Oracle Universal Installer, the configuration information for the corresponding application server instance is not automatically removed from Oracle Internet Directory. For example, referring to [Figure 12-1, "Oracle9iAS Tree in Oracle Internet Directory"](#) on page 12-6, if you deinstall the application server installation `IAS2` using Oracle Universal Installer, the node named `IAS2` is not automatically removed from under `IAS Instance` in the tree.

It is not a problem to have unused instances in Oracle Internet Directory unless you would like to reuse the same instance name for a subsequent installation. In this case, you must manually remove the instance from Oracle Internet directory before installing the new instance. For example, if you deinstall the `IAS2` application server installation using Oracle Universal Installer and would like to install a new installation called `IAS2`, you must remove the `IAS2` instance node from Oracle Internet Directory before doing the new installation.

To remove an application server instance from Oracle Internet Directory:

1. Start Oracle Directory Manager:

(UNIX) `ORACLE_HOME/bin/oidadmin`

(Windows) From the Start menu, select Programs -> `ORACLE_HOME` -> Integrated Management Tools -> Oracle Directory Manager

2. Log in to Oracle Directory Manager as `cn=orcladmin` or a user that is a member of the `IASAdmins` group.
3. Expand the Entry Management entry in the System Objects frame by clicking the plus sign (+) next to it.
4. Expand `cn=OracleContext`.
5. Expand `cn=Products`.
6. Expand `cn=IAS`.
7. Expand `cn=IAS Instances`.

8. Under `cn=IAS Instances`, select the instance you would like to remove.
9. In the toolbar, select the Edit menu and select Delete.
10. In the Systems Object frame, under `cn=OracleContext`, expand `cn=Groups`.
11. Select `cn=IASAdmins`.
12. In the Properties tab, remove the instance from the `uniqueMember` field. **Do not select Edit/Delete from the toolbar**, rather, edit the `uniqueMember` field with the cursor to remove the instance. Click **Apply**.

13

High Availability

This chapter provides information about high availability within Oracle9*i* Application Server.

It contains the following topics:

- [Introduction to High Availability](#)
- [Oracle Process Management and Notification](#)
- [Configuring the Oracle Process Manager](#)
- [Configuring the Oracle Notification System](#)
- [Failure Scenarios](#)

Introduction to High Availability

The availability of a system or any component in that system is defined by the percentage of time that it works normally. A system works normally when it meets its correctness and performance specifications. For example, a system that works normally for twelve hours per day is 50% available. A system that has 99% availability is down 3.65 days per year on average. System administrators can expect critical systems to have 99.99% or even 99.999% availability. This means that the systems experience as little as four to five minutes of downtime per year.

Availability may not be constant over time. For example, availability may be higher during the daytime when most transactions occur, and lower during the night and on weekends. However, because the Internet provides a global set of users, it is a common requirement that systems are always available.

Redundant components can improve availability, but only if a spare component takes over immediately for a failed component. If it takes ten minutes to detect a component failure and twenty additional minutes to start the spare component, then the system experiences a 50% reduction in availability for that hour of service.

Oracle9iAS is designed to provide maximum system availability during many types of hardware and software failures.

Benefits of High Availability

Oracle9i Application Server keeps your system available by providing the following benefits:

- **Fault Tolerance:** Oracle9iAS has a flexible deployment model so you can deploy an architecture that exposes no single point of failure. This means that despite the failure of any component in the system, the application server is tolerant to failures and continues to function and service client requests.
- **Death Detection and Restart:** To have maximum availability, when a server process fails, the system should take appropriate action by cleaning up memory and restarting the failed processes. The management processes in each Oracle9iAS instance monitor the HTTP server and OC4J processes and restart failed processes as required.
- **Failover:** Failover hides system failure from users. Failures do occur, but clients should not perceive any loss of service as their requests are failed-over/re-routed to another redundant service that takes over for the failed one.

At the mod_oc4j-OC4J level, after the failure of a stateless service, the application server routes requests to alternate instances of the service in a similar fashion to connection rerouting. After a failure of a stateful service, the application server reroutes the request to an alternate instance to which the state has been replicated.

At application server instance level, you can use a hardware load balancer, Oracle9iAS Web Cache, or a DNS round robin strategy to load balance requests across redundant application server instances. This allows your Web site to continue functioning even if one of the application server instances goes down.

See Also: For instructions on configuring Oracle9iAS Web Cache as a load balancer to route requests, refer to *Oracle9iAS Web Cache Administration and Deployment Guide*.

- **Connection Rerouting:** Rerouting occurs when a process or node servicing a request fails while a session is still executing. To address this failure, client requests to the session are transparently rerouted to alternate application server instances in the cluster.

High Availability with Oracle9iAS Web Cache

Oracle9iAS Web Cache supports the following content-aware load balancing and failover detection features:

- **Load Balancing:** Most Web sites are served by multiple origin servers running on multiple computers that share the load of HTTP and HTTPS requests. This feature enables Web sites to be built with a collection of servers for better scalability and reliability. Oracle9iAS Web Cache sends requests to the origin server with the most available load.
- **Failover:** After a specified number of continuous request failures, Oracle9iAS Web Cache considers an origin server as failed. When an origin server fails, Oracle9iAS Web Cache automatically distributes the load over the remaining origin servers based on the available load. Oracle9iAS Web Cache polls the failed origin server for its current up/down status until it is back online. When the failed server returns to operation, Oracle9iAS Web Cache includes its available load to load balance requests.

- **Surge Protection:** To prevent an overload of requests on the origin servers, Oracle9iAS Web Cache provides a surge protection feature that enables you to set a limit on the number of concurrent requests to the origin servers. When the limit is reached, subsequent requests are queued to wait up to a maximum amount of time. If the maximum wait time is exceeded, Oracle9iAS Web Cache rejects the request and serves a site busy apology page to the Web browser that initiated the request.
- **Cache Clusters:** With cache clusters, cache content is replicated among multiple Oracle9iAS Web Cache servers. If a cluster member fails, other members of the cache cluster detect the failure and take over ownership of the cacheable content of the failed cluster member. Because requested cacheable documents are likely to be stored in the cache of surviving cluster members, fewer requests for cacheable documents need to be routed to the Web server.

See Also: [Chapter 14, "Application Server Clustering"](#)

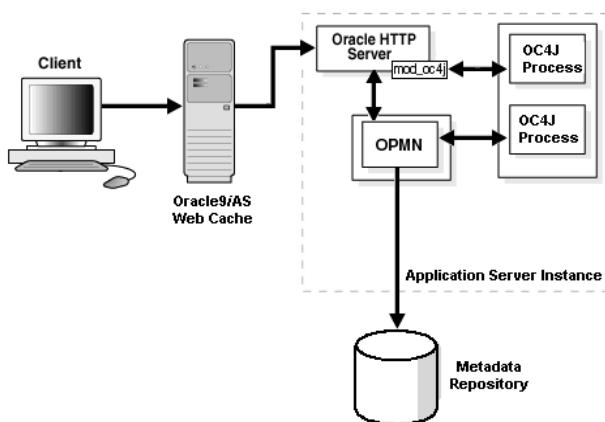
- **Auto-Restart:** To handle cache failures, Oracle9iAS Web Cache provides an auto-restart process that automatically restarts the cache.
- **Dynamically Generated Content:** By caching dynamically-generated content, Oracle9iAS Web Cache adds a buffer between the user load and the origin server. As a result, if you configure Oracle9iAS Web Cache to cache and assemble some of the dynamically generated content, then Oracle9iAS Web Cache takes the burden off the origin servers and databases.

For instance, certain viruses or attacks attempt to bring down sites by flooding servers with malicious requests, some of which are intended to exploit weaknesses in the application Web server. Because attacks are typically distributed, it is often difficult to block all the IPs that are sending the requests. However, by caching the responses to the initial requests, Oracle9iAS Web Cache can shield the backend servers from having to respond to subsequent requests of this nature. In other words, caching provides some level of denial-of-service protection.

Architecture

Figure 13–1 shows an example of high availability architecture where an Oracle9iAS instance can contain one Oracle HTTP Server instance and zero or more OC4J processes. Here, Oracle9iAS Web Cache is used as a load balancer to route requests. OPMN within the Oracle9iAS instance environment monitors the Oracle HTTP Server and OC4J processes. If either Oracle HTTP Server or OC4J fail, the OPMN restarts the failed process. This enables a maximum level of availability of the components within an Oracle9iAS instance.

Figure 13–1 High Availability Architecture



The `mod_oc4j` Module

The `mod_oc4j` module routes requests from the Oracle HTTP Server to Oracle Containers for J2EE (OC4J). It works in conjunction with OPMN to keep its routing table updated so that it load balances across only live OC4J processes in an OC4J instance. When a request comes in, `mod_oc4j` selects a process from its shared memory table. If a session is created with this request, related requests are routed to the same process. Otherwise, a new process is selected based on the load balancing algorithm. To allow fault tolerance, this module supports OC4J instances and islands. To handle failover with stateless requests, it tries to select an alternative OC4J process within the same island and routes the request to that process. If all the OC4J processes in the instance have been tried and none of them are successful, `mod_oc4j` returns request failure. For session enabled requests, the module tries to failover to another OC4J process within the same island as the original process that

serviced the request. Again, if all the OC4J processes in the island are unable to service the request, `mod_oc4j` returns request failure.

See Also: *Oracle HTTP Server Administration Guide*

Oracle Process Management and Notification

Oracle Process Management and Notification Service (OPMN) manages the processes within an Oracle9iAS instance and also channels all notifications from different components instances to all interested in receiving them.

OPMN consists of the following two components:

- [Oracle Process Manager](#)
- [Oracle Notification System](#)

Oracle Process Manager

Oracle Process Manager (PM) is the centralized process management mechanism in Oracle9iAS and is used to manage all Oracle HTTP Server and OC4J related processes. It starts, stops, restarts, and detects death of these processes. When these processes are configured to start up, the characteristics of each set is specified in a configuration file called `opmn.xml`. The Oracle Enterprise Manager Web site also uses PM to manage processes.

The PM starts and then waits for a command to start specific or all processes. Similarly, at shutdown, the PM receives a request to stop one or more processes, or all processes and itself as specified by the HTTP request parameters.

Oracle Notification System

Oracle Notification System (ONS) is the transport mechanism for failure, recovery, startup, and other related notifications between components in Oracle9iAS. It operates according to a subscriber-publisher model, wherein any component that wishes to receive a notification of a certain type subscribes to the ONS. When such a notification is published, ONS sends it to all subscribers. PM and ONS exist within the same process, and comprise Oracle Process Management and Notification. OPMN is monitored by a shadow process that restarted upon request or after a catastrophic failure

Configuring the Oracle Process Manager

The Oracle Process Manager uses a configuration file, `opmn.xml`, that contains information about all the processes it manages. This information includes parameters to start a process as well as metadata for bookkeeping. Each process type that OPMN manages has a corresponding element defined in `opmn.xml`. This section discusses the following elements:

- [process-manager](#)
- [ohs](#)
- [oc4j](#)
- [custom](#)

process-manager

This is the topmost element for the process manager. The administrative process defines a single process manager. A `process-manager` element contains elements of types:

- [ohs](#), [oc4j](#), [custom](#): Specify managed processes.
- [log-file](#): Specifies the location of the process manager log file. It is optional. The default is `ORACLE_HOME/opmn/logs/ipm.log`.

The `log-file` element can have one attribute: `level` (optional). The `level` attribute can be one of 1=FATAL, 2=ERROR, 3=WARN, 4=NOTIFY, 5=DEBUG, 6=VERBOSE. The default is 3 (warning, error, and fatal messages are logged).

ohs

This element defines an Oracle HTTP Server process and has the following attributes:

- **gid**: A unique identifier for the process. It is optional. The default is "HTTP Server".
- **maxRetry**: Specifies the number of times PM should check the Oracle HTTP Server process before declaring it dead, and the number of times to try to start an Oracle HTTP Server process before giving up. It is optional. The default value is 3.

- **timeout**: Specifies the maximum time needed to wait for Oracle HTTP Server to completely start. It is optional. The default value is dependent on system load, but is at least 300 seconds.

The `ohs` element can contain optional elements of the following types:

config-file: Specifies the server configuration file. It is optional. The default value is `ORACLE_HOME/Apache/Apache/conf/httpd.conf`.

The `config-file` element can have these attributes:

- **path**: The path to the configuration file. If you specify a relative path, then the current Oracle HTTP Server working directory is prepended to the path you specify.
- **start-mode**: Specifies the mode in which the server starts: `ssl` or `non-ssl`. It is optional. The default is `non-ssl`.

The `start-mode` element can have the `mode` attribute. Specify `ssl` to start Oracle HTTP Server in SSL mode.

oc4j

This element defines an OC4J process group and has the following attributes:

- **gid**: Unique identifier for the OC4J process group. It is optional. The default is `home`.
- **numProcs**: Number of OC4J processes to start in this OC4J process. It is optional. This value is ignored if the `island` element is specified for this OC4J process. The default is 1.
- **maxRetry**: Number of times PM checks OC4J before declaring it dead, and the number of times to try to start it before giving up. It is optional. The default is 3.
- **instanceName**: Defines a group of OC4J processes, or islands, that service the same set of applications. It is optional. The default is `home`.
- **timeout**: Specifies the maximum time needed to wait for Oracle HTTP Server to completely start. It is optional. Default value is dependent on system load, but is at least 300 seconds.

The `oc4j` element can contain elements of the following types:

- **config-file**: Typically, specifies the `server.xml` file if it is in a non-default location. It is required. If a relative path is specified, the current OC4J working directory is prepended to the path.
- **java-bin**: Specifies the path to the Java executable that starts OC4J. It is optional. The default is `ORACLE_HOME/jdk/bin/java`.
- **java-option**: Specifies JVM options required by the OC4J command line. It is optional.
- **oc4j-option**: Specifies OC4J options required by the OC4J command line. It is optional.
- **port**: Specifies the individual port or range of ports on which OC4J listens. It is optional if `numProcs` has not been specified or is equal to 1, required if `numProcs` is greater than 1.

The `port` element requires the following attributes:

AJP: AJP ports on which the OC4J process listens. If the value is set to 0, then OC4J picks the AJP port.

JMS: JMS ports on which the OC4J process listens.

RMI: RMI ports on which the OC4J process listens.

For example:

```
<port ajp="8000,8010-8020,8003" jms="7030-7050,7780-7790"  
rmi="9000,9001,9002,9003,9004" />
```

- **island**: A set of JVMs sharing session state within an OC4J process. For example, if a session is being handled by OC4J process 1 in island 1 of specifying this element and process 1 dies, `mod_oc4j` fails the session over to OC4J process 2, which is also a member of island 1.

To start multiple islands within an OC4J process, you must specify an `island` element for each island. In this case the `numProcs` attribute, if specified, for OC4J is ignored. If you want to start only one island within an OC4J process, you can specify `numProcs` either as an OC4J attribute or as an `island` attribute.

The island element requires the following attributes:

- **id**: Unique identifier for the island.
- **numProcs**: Number of processes to start for this island and OC4J process.

For example:

```
<island id="island1" numProcs="2"/>
<island id="island2" numProcs="3"/>
```

The environment element can have the following element type:

prop: Represents one environment name value pair for the custom process.

The prop element can have the following attributes:

- **name**: Name of the environment name value pair
- **value**: Value of the environment name value pair

Example OC4J Configurations:

This section contains examples of OC4J configurations.

Start Four JVMs in the Same Island in One OC4J Process

```
<oc4j instanceName="oc4jInstance1">
  <config-file path="/oracle/j2ee/home/config/server.xml" />
  <port ajp="6666-6670" rmi="6686-6690" jms="6786-6790"/>
  <island id="island" numProcs="4"/>
</oc4j>
```

or

```
<oc4j instanceName="oc4jInstance1" numProcs="4">
  <config-file path="/oracle/j2ee/home/config/server.xml" />
  <port ajp="6666-6670" rmi="6686-6690" jms="6786-6790"/>
</oc4j>
```

These two configurations achieve the same results except that the second one uses `default_island` as its OC4J island ID.

Start Four JVMs (Two in Island 1, Two in Island 2) in One OC4J Process <

```
<oc4j instanceName="oc4jInstance1">
  <config-file path=/oracle/j2ee/home/config/server.xml />
  <port ajp="6666-6670" rmi="6686-6690" jms="6786-6790"/>
  <island id="island1" numProcs="2"/>
  <island id="island2" numProcs="2"/>
</oc4j>
```

or

```
<oc4j instanceName="oc4jInstance1" numProcs="4">
  <config-file path=/oracle/j2ee/home/config/server.xml />
  <port ajp="6666-6670" rmi="6686-6690" jms="6786-6790"/>
  <island id="island1" numProcs="2"/>
  <island id="island2" numProcs="2"/>
</oc4j>
```

These two configurations achieve the same result. Since the island element is specified, the OC4J attribute numProcs is ignored.

custom

This element defines a generic process group and has the following attributes:

- **gid**: Unique identifier for this custom process group. It is optional. The default is default_gid1.
- **num_of_proc**: Number of processes to start in this group. It is optional. The default is 1.
- **maxRetry**: Specifies the number of times to try to start the process before giving up. It is optional. The default is 3.

The **custom** element can contain elements of type:

- **start**: Specifies the absolute path to the command used to start the process It is required.
- **environment**: Specifies the environment required by the process when it is spawned.

The environment element can have one of the following element types:

prop: Represents one environment name value pair for the custom process.

The prop element can have the following attributes:

- **name**: Name of the environment name value pair
- **value**: Value of the environment name value pair

Sample Configurations for Oracle Process Manager

This section provides some sample configurations for the Oracle Process Manager. Electronic copies of these are available at

<http://otn.oracle.com>

One Oracle HTTP Server Process, One OC4J Process with Defaults

This configuration starts one Oracle HTTP Server and one OC4J process with default values.

```
<notification-server>
  <port local="6001"
        remote="6002">
    </port>
    <log-file path="/private/my_directory/tmp/opmn_logs/ons.log"
              level="5">
      </log-file>
</notification-server>
<process-manager>
  <ohs/>
  <oc4j>
    <config-file path=ORACLE_HOME/j2ee/home/config/server.xml />
  </oc4j>
  <log-file path="/private/my_directory/tmp/opmn_logs/ipm.log"
            level="4">
    </log-file>
</process-manager>
```

One Oracle HTTP Server Process, Two OC4J Processes, and One Generic Process with User-Specified Values

This configuration starts one Oracle HTTP Server process, two OC4J processes, and one generic process with several user-specified values.

```
<process-manager>
  <ohs gid="a1" maxRetry="3">
    <config-file path="/my_directory/conf/httpd.conf"/>
  </ohs>
  <oc4j gid="o1" numProcs="2" maxRetry="4" clusterID="myClusterA"
        islandID="myIslandA">
    <config-file path="/my_directory/conf/oc4j.xml"/>
      <port http="8401" rmi="8600" jms="6577" ajp="3456"/>
    </oc4j>
    <custom gid="g1" num_of_proc = "1">
      <start path="/my_directory/bin/exec1"/>
      <environment>
        <prop name="PATH" value="/my_directory/ias/lib"/>
        <prop name="CLASSPATH" value="/my_directory/ias/bin">
      </environment>
    </general>
  </process-manager>
```

One Oracle HTTP Server Process, Two OC4J Processes with User-Specified Values

This configuration starts one Oracle HTTP Server and two OC4J processes with some user-specified values.

```
<notification-server>
  <port local="6001"
        remote="6002">
    </port>
    <log-file path="/private/my_directory/tmp/opmn_logs/ons.log"
              level="5">
    </log-file>
</notification-server>
<process-manager>

<!-- Start one ohs process with a process group ID of a1, with a config file in
a non-default directory location -- >

<ohs gid="a1" maxRetry="3">
  <config-file path="Apache/Apache/my_conf/httpd.conf"/>
</ohs>

<!-- Start two oc4j processes with a process group ID of o1, with a instanceName
of myClusterA (This should be the worker specified
in
the mod_oc4j config file), and an islandID of myIslandA. Since we are starting
two processes we have to specify the port range used when starting the
processes. So OC4J process1 starts with ajp port 8010, jms port 8020
and rmi port 8030. The second OC4J process uses ajp port 8011, jms port 8021,
and rmi port 8031. -->

<oc4j gid="o1" numProcs="2" instanceName="myInstanceA"
           islandID="myIslandA">
  <port ajp="8010-8012", jms="8020-8022" rmi="8030-8032"/>
</oc4j>

<!-- If the logs should be created in a specific directory provide that location
here --!>

<log-file path="/tmp/ipm.log" level="4"/>
</process-manager>
```

Configuring the Oracle Notification System

The Oracle Notification System (ONS) uses a configuration file, `opmn.xml`. This section discusses the following elements in `opmn.xml`, under the “notification-server” section:

- [port](#)
- [ssl](#)
- [log-file](#)

port

This specifies the ports to which ONS binds and listens for notifications or the start, stop, restart, and statistical dump requests.

Attributes

- **local**: Bound to “localhost”, this port is required and is used by all ONS clients on this host (including Oracle HTTP Server and OC4J).
- **remote**: Bound to the current host, this optional port is used for communication with other OPMN servers.
- **request**: Bound to the current host, this optional port is used for statistical dump requests only. This port is required by the DCM layer, which is the layer about OPMN.

ssl

This specifies SSL information for the ONS remote port. It is optional.

Attributes

- **enabled**: Set to “true” if ssl is enabled, otherwise “false”. It is required.
- **wallet-file**: Specifies the path of the wallet file to use. It is optional.
- **wallet-password**: Specifies the password for the given wallet-file. It is required if wallet-file is specified.

log-file

This specifies the location of the notification manager log file. It is optional. Default is *ORACLE_HOME*/opmn/logs/ons.log.

The `log-file` element can have one attribute: `level`. It is optional. The `level` attribute can be one of 1=FATAL, 2=ERROR, 3=WARN, 4=NOTIFY, 5=DEBUG, 6=VERBOSE. The default is 3 (warning, error, and fatal messages are logged).

Sample Configuration Examples for Oracle Notification System

This section provides some sample configurations for the Oracle Notification System.

Example 1

```
<notification-server>
<port local ="6100" request="6300"/>
<log-file path="/ora/opmn/logs/ons.log" level="4"/>
</notification-server>
```

Example 2

```
<notification-server>
<port local ="6100" remote="6200" request="6300"/>
<ssl enabled="true" wallet-file="/ora/wal" wallet-password="foo">
<log-file path="/ora/opmn/logs/ons.log" level="4"/>
</notification-server>
```

Failure Scenarios

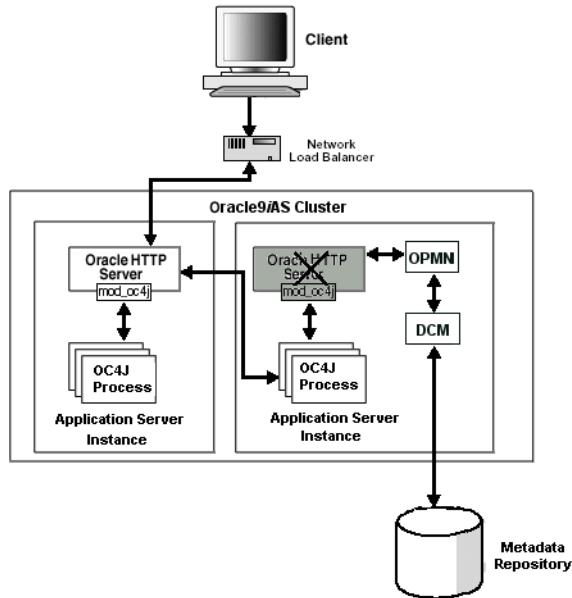
This section discusses the common failure scenarios and explains the response from the high availability infrastructure in Oracle9i Application Server. Topics include:

- [Oracle HTTP Server Goes Down](#)
- [OC4J Process Goes Down](#)
- [Hi-AV Framework \(OPMN\) Goes Down](#)

Oracle HTTP Server Goes Down

[Figure 13–2](#) depicts a scenario where Oracle HTTP Server goes down.

Figure 13–2 Oracle HTTP Server Goes Down



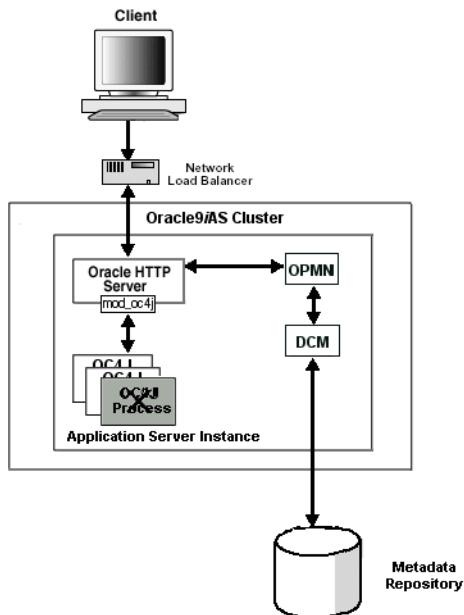
- **Death Detection:** The external systems, hardware load balancers or Oracle9iAS Web Cache, detect this and do not route requests to the dead Oracle HTTP Server, but instead route to the live one. OPMN is also notified of this event through one of its missed heartbeats, or from the OS, since it monitors that process through its process identifier.

- **Restart:** OPMN restarts the dead Oracle HTTP Server automatically. No external intervention is required. Oracle HTTP Server automatically retrieves the updated configuration file through DCM. In general, the restart is so quick that there should not have been any major configuration changes in the brief time frame.
- **Fault Impact:** The Oracle HTTP Server connection to OC4J is terminated. However, the session is not lost. All requests in process continue through the live Oracle HTTP Server.
- **States Recovered:** When Oracle HTTP Server comes back up, it rebuilds its routing information to all OC4J processes in the cluster and resumes steady state almost instantly. For non-J2EE applications, there is an additional impact of rebuilding all database connections and the like. Note that the J2EE session information is kept in the OC4J processes and is not impacted by Oracle HTTP Server being down.

OC4J Process Goes Down

[Figure 13–3](#) depicts a scenario where OC4J goes down.

Figure 13–3 OC4J Goes Down



- **Death Detection:** OPMN is informed through one of the missed heartbeats or from the OS, since it monitors that process through its process identifier.
- **Restart:** OPMN restarts OC4J automatically. No external intervention is required. OC4J automatically retrieves the updated configuration, including new applications deployed, through DCM. However, in general, the restart is so quick that there should not have been any major changes in the brief time window.
- **Fault Impact:** The impact on a J2EE application is minimal. The session information is distributed to other OC4J processes in an OC4J cluster. Through OPMN, all Oracle HTTP Server routing tables are updated to reflect the downed OC4J. No further J2EE requests are sent to that OC4J process. There might be some performance impact since there are fewer OC4J processes running in the system. However, most systems should be able to handle this.

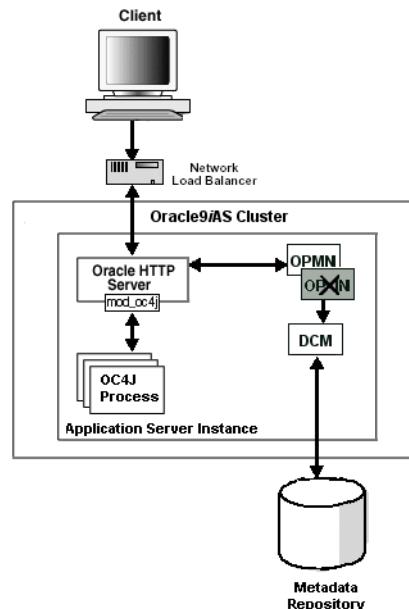
- For non-J2EE applications, such as PL/SQL or Perl, there is no impact at all.
- **States Recovered:** There is no impact on existing session states, since they are replicated across all OC4J processes within an island. The request is routed to another process in the same island in the same cluster. OC4J process automatically receives the updated configuration, including new application deployed, through DCM. OPMN updates the Oracle HTTP Server routing tables to reflect the re-addition of the new process, so routing to it resumes.

Hi-AV Framework (OPMN) Goes Down

Since OPMN does not play a role in actively servicing requests, this is not a critical failure. However, since it is critical in ensuring system-wide updates, its failure recovery is important to ensuring a healthy steady state.

[Figure 13–4](#) depicts a scenario where OPMN goes down.

Figure 13–4 OPMN Goes Down



- **Death Detection:** A shadow process monitors OPMN. No other processes in the system monitor it.

- **Restart:** The shadow process that continually watches over OPMN ensures it comes back in case of a failure.
- **Fault Impact:** The impact is minimal. The routing information that OPMN communicates to the critical modules, such as mod_oc4j, might get stale while OPMN is down, resulting in less optimal routing.
- **States Recovered:** All routing table information and the past events are recovered. The regular heartbeats — ping and reverse ping — help the restarted OPMN re-build the cluster wide map in its memory.

Application Server Clustering

This chapter describes the concept of application server clustering and provides instructions on how to create and manage clusters.

It contains the following topics:

- [About Oracle9iAS Clustering](#)
- [Architecture](#)
- [Oracle Enterprise Manager Configuration Tree](#)
- [Instance-Specific Parameters](#)
- [Examples](#)
- [Cluster Configuration](#)

About Oracle9iAS Clustering

A cluster is a set of application server instances configured to act in concert to deliver greater scalability and availability than a single instance can provide. While a single application server instance can only leverage the operating resources of a single host, a cluster can span multiple hosts, distributing application execution over a greater number of CPUs. While a single application server instance is vulnerable to the failure of its host and operating system, a cluster continues to function despite the loss of an operating system or host, hiding any such failure from clients.

Clusters leverage the combined power and reliability of multiple application server instances while maintaining the simplicity of a single application server instance. For example, browser clients of applications running in a cluster interact with the application as if it were running on a single server. The client has no knowledge of whether the application is running on a single application server or in an application server cluster. From the management perspective, an application server administrator can perform operations on a cluster as if the administrator was interacting with a single server. An administrator can deploy an application to an individual server; the application is propagated automatically to all application server instances in the cluster.

The following sections discuss how application server clustering increases scalability, availability, and manageability.

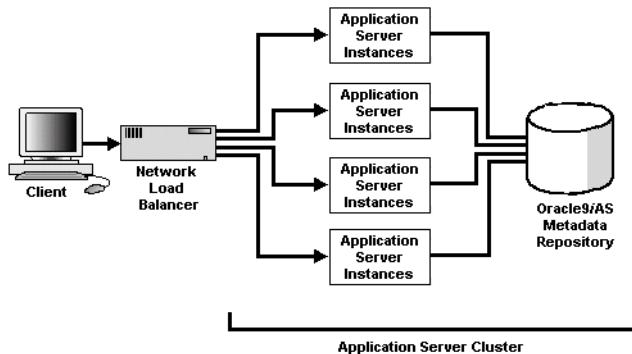
- [Scalability](#)
- [Availability](#)
- [Manageability](#)
- [Component Support](#)
- [Non-Managed Clustering](#)

Scalability

Oracle9iAS clustering enables you to scale your system beyond the limitations of a single application server instance on a single host. [Figure 14-1](#) shows how a cluster unifies multiple application server instances spread over multiple hosts to collectively serve a single group of applications. In this way, clustering makes it possible to serve increasing numbers of concurrent users after the capacity of a single piece of hardware is exhausted.

Clients interact with the cluster as if they are interacting with a single application server. An administrator can add an application server instance to the cluster while the cluster is operating, increasing system capacity without incurring downtime.

Figure 14–1 Oracle9iAS Cluster



Clients access the cluster through a load balancer, which hides the application server configuration. The load balancer can send requests to any application server instance in the cluster, as any instance can service any request. An administrator can raise the capacity of the system by introducing additional application server instances to the cluster, each of which derives its configuration from a shared Oracle9iAS Metadata Repository.

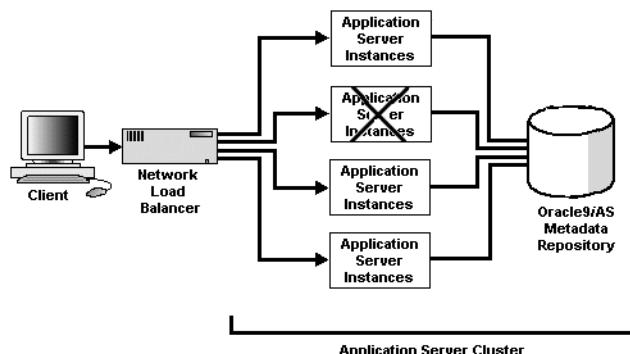
Availability

Oracle9iAS clustering enables you to achieve a higher level of system availability than is possible with only a single application server instance. An application running on a single instance of an application server is dependent on the health of the operating system and host on which the server is running. In this case, the host poses as a single point of failure because if the host goes down, the application becomes unavailable.

An application server cluster eliminates the single point of failure by introducing redundancy and failover into the system. Any application server instance in the cluster can service any client request, and the failure of any single instance or host does not bring down the system. Client session state is replicated throughout the cluster, thereby protecting against the loss of session state in case of

process failure. The administrator can configure the extent of session state replication.

Figure 14–2 Application Server Instance Failure in a Cluster



[Figure 14–2](#) illustrates how application server clusters enable higher availability by providing redundancy and backup and eliminating a single point of failure. Clients access the cluster through a load balancer which can send requests to any application server instance in the cluster. In the case that an application server instance becomes unavailable, the load balancer can continue forwarding requests to the remaining application server instances, as any instance can service any request.

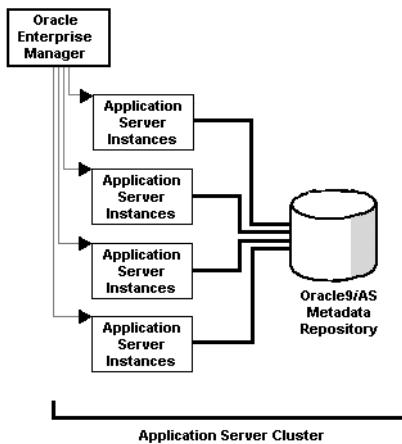
Manageability

[Figure 14–3](#) demonstrates how managed clustering uses Oracle Enterprise Manager. While any clustered system requires all instances to be similarly configured in order to function properly, Oracle9iAS managed clustered instances synchronize their configurations automatically, relieving the administrator of the responsibility to manually update each individual instance. Using Oracle Enterprise Manager, the administrator can make configuration changes as if on a single application server instance. Applicable changes are propagated automatically to all instances in the cluster.

Oracle9iAS cluster management simplifies the tasks of creating and administering clusters and reduces the chance of human error corrupting the system. An administrator creates a cluster in a single management operation. Then, the administrator adds the initial application server instance to the cluster to define the

base configuration for the cluster. Any additional instances automatically inherit this base configuration.

Figure 14–3 Oracle Enterprise Manager Manages a Cluster



Component Support

Oracle9iAS clustering applies to the synchronization and management of Oracle HTTP Server (OHS) and Oracle9iAS Containers for J2EE (OC4J) components.

Other Oracle9iAS components, such as Oracle9iAS Web Cache, may support a component-specific clustering model or cluster-like functionality. This is separate from application server clustering and is not discussed in this chapter. Please see the component documentation for further details. For more information about Oracle9iAS Web Cache clustering, see *Oracle9iAS Web Cache Administration and Deployment Guide*.

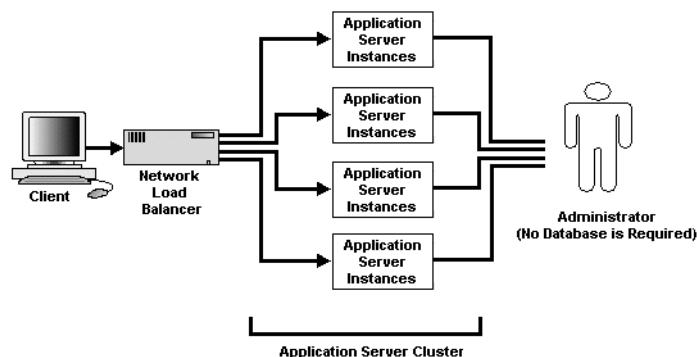
Non-Managed Clustering

This chapter discusses managed application server clusters that offer scalability, availability, and manageability. Managed application server clusters require a metadata repository to store shared configuration data.

Oracle9iAS also enables you to create non-managed application server clusters that do not require a metadata repository and therefore have no database dependency. Non-managed clusters provide scalability and availability, but *not* manageability. In

a non-managed cluster, it is your responsibility to synchronize the configuration of the application server instances. [Figure 14–4](#) illustrates that a non-managed cluster does not require a database, but you have to configure each application server instance yourself.

Figure 14–4 Non-Managed Clustering



If you want to cluster J2EE applications and do not want to use a metadata repository, there are two types of non-managed clusters that you can use:

- Non-managed application server cluster
- OC4J-only cluster

Non-Managed Application Server Cluster

Create a non-managed application server cluster if you want to use both OHS and OC4J. In a non-managed application server cluster, `mod_oc4j` will load-balance requests to all OC4J instances in the cluster.

For more information on non-managed application server clustering, see the Oracle9iAS page on OTN at <http://otn.oracle.com/products/ias>.

OC4J-Only Cluster

Create an OC4J-only cluster if you want to use the standalone OC4J that is available for download from OTN. In an OC4J-only cluster, the Java load balancer load-balances requests to all OC4J instances in the cluster. An OC4J-only cluster has

a lightweight disk footprint, but the Java load balancer can be a single point of failure.

For more information on OC4J-only clustering, see the OC4J page on OTN at <http://otn.oracle.com/tech/java/oc4j>.

Architecture

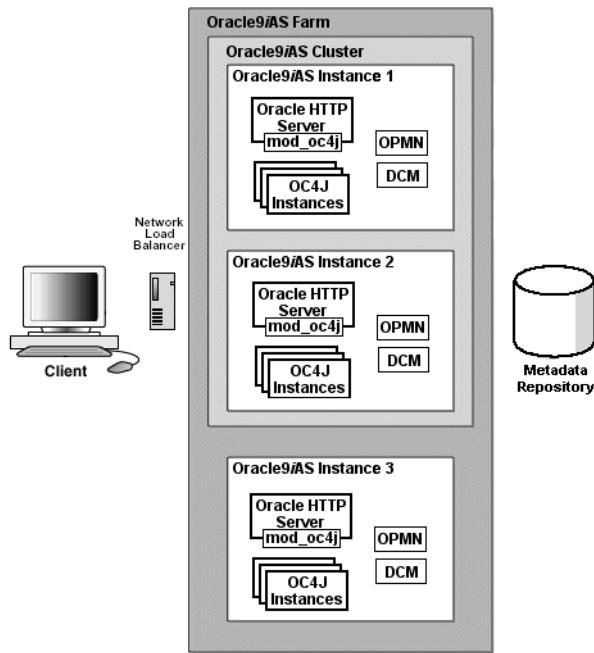
A cluster coordinates several application server instances and its components. The roles of the components included in the cluster are described in the following sections:

- [Front-End Load Balancer](#)
- [Metadata Repository in the Infrastructure](#)
- [Farm](#)
- [Cluster](#)
- [Application Server Instance](#)
- [Management Features](#)
- [Component Instances](#)
- [J2EE Applications](#)

[Figure 14–5](#) shows the architecture of a farm and a cluster. There are three application server instances, where each instance shares the same Oracle9iAS Metadata Repository within an infrastructure. Thus, all three application server instances are part of the same farm.

Application server instances 1 and 2 are involved in a cluster together. In front of the cluster is a front-end load balancer. Included within each application server instance are its manageability features—Oracle Process Management and Notification (OPMN) and Dynamic Configuration Management (DCM)—and its installed components—Oracle HTTP Server and Oracle9iAS Containers for J2EE (OC4J).

Figure 14–5 Oracle9iAS Cluster Architecture



Front-End Load Balancer

After you have created a cluster, you can add a load balancer in front of all application server instances in the cluster, which provides availability and scalability for the application server instances.

We recommend that you purchase and install a hardware load balancer for the best performance. Alternatively, you could use Web Cache as a load balancer, which could be a single point of failure.

See Also: *Oracle9iAS Web Cache Administration and Deployment Guide* for instructions on how to set up Oracle9iAS Web Cache as your load balancer for your cluster

Metadata Repository in the Infrastructure

When you install Oracle9iAS, you have the option of installing the Oracle9iAS Infrastructure. An Oracle9iAS Infrastructure provides Oracle Internet Directory, Oracle9iAS Single Sign-On, and the Oracle9iAS Metadata Repository. The metadata repository is an Oracle9i database that is used to store the application server instance information and configuration. The application server instance tables are created in the metadata repository. Multiple application server instances can share the metadata repository of the infrastructure.

Application server instances associate with an infrastructure either during installation or through the Oracle Enterprise Manager after installation.

Farm

A farm is a group of multiple application server instances that associate with the same metadata repository. The application server instances that belong to a farm can be installed anywhere on the network.

- It is only within the constraint of a farm that you can create a cluster.
- A farm can host multiple clusters.

Cluster

A cluster is a logical group of application server instances that belong to the same farm. Each application server instance may be part of only one cluster. If an instance is part of a cluster, then all of its configured components are implicitly part of that cluster. Each application server instance can only be configured with Oracle HTTP Server and OC4J components to be contained in a cluster. A cluster can include zero or more application server instances.

All application server instances involved in the cluster have the same “cluster-wide” configuration. If you modify the configuration on one application server instance, then the modification is automatically propagated across all instances in the cluster.

Note: “Instance-specific” configuration parameter modifications are not propagated. For a description of these parameters, see [“Cluster Configuration”](#) on page 14-26.

Application Server Instance

An application server instance consists of a single Oracle HTTP Server and one or more OC4J instances. It is a single installation in one Oracle home. If you have multiple application servers on a single host, each is installed into its own Oracle home and uses separate port numbers.

To manage clusters from Oracle Enterprise Manager, the application server uses a metadata repository for storing its tables and configuration. Each application server instance in the cluster has the same base configuration. The base configuration contains the cluster-wide parameters and excludes instance-specific configuration. If you modify any part of the cluster-wide configuration, the modifications are propagated to all other application server instances in the cluster. If you modify an instance-specific parameter, it is not propagated, as it is only applicable to the specified application server instance. See "["Cluster Configuration"](#)" on page 14-26 for a listing of the instance-specific parameters. The cluster-wide parameters are all other parameters.

In order for each application server instance to be a part of a cluster, the following must be true:

- The application server instances you add to a cluster must be part of the same farm as the cluster, which means they all use the same metadata repository. You can associate application server instances with the same metadata repository during install time or after installation using Oracle Enterprise Manager.
- The first application server instance you add to the cluster must contain only OC4J and Oracle HTTP Server components. Web Cache can be configured, but it will be ignored for clustering operations. If other Oracle9iAS components are part of the application server instance, Oracle9iAS displays an error and does not add the application server instance to the cluster.

Note: Oracle9iAS Web Cache provides its own clustering functionality separate from application server clustering. See *Oracle9iAS Web Cache Administration and Deployment Guide* for more information.

- When you install additional application server instances, ensure that only Oracle HTTP Server, OC4J, and Oracle9iAS Web Cache are configured. The Oracle9iAS Web Cache will be ignored for clustering operations.
- Each application server instance can contain only one Oracle HTTP Server.

- Each application server instance can contain one or more OC4J instances.
- Each application server instance must reside on a host using the same operating system type (for example, UNIX) as the other instances in the cluster.

To cluster application server instances, do the following:

1. Create an empty cluster in the farm. The only requirement for creating a cluster is a unique name.
2. Add the first application server instance to the cluster. This application server instance must already belong to the farm. The configuration of this first instance is used as the base configuration for all additional application server instances. The base configuration overwrites any existing configuration of subsequent application server instances that join the cluster.
The base configuration includes the cluster-wide properties. It does not include instance-specific properties. See "["Cluster Configuration"](#)" on page 14-26 for more information about instance-specific properties.
3. Add other application server instances—even if they exist on another host—to the cluster. Each additional application server instance inherits the base configuration.
4. If you add application server instances into a cluster, set the base configuration, then remove all application server instances from a cluster, the cluster is then empty and the base configuration is not set. Thus, the next application server instance that you add becomes the source of the base configuration.
5. When added to or removed from the cluster, the application server instance is stopped. You can restart the added application server instances within the context of the cluster. You can restart the removed application server instance from the standalone instances section in the farm.

Once grouped in the same cluster, these application server instances have the following properties:

- Each application server instance has the same cluster-wide configuration. That is, if you modify any cluster-wide parameters, the modifications are propagated to all application server instances in the cluster. For instance-specific parameters, you must modify these on each individual application server instance.
- If you deploy an application to one application server instance, it is propagated to all application server instances in the cluster. The application is actually deployed to an OC4J Instance in the application server instance and propagated to the same OC4J Instance in the other application server instances in the

cluster. You can change the configuration for the deployed application, and this change is propagated to the same OC4J Instance in the other application server instances in the cluster.

- Each application server instance is equal in the cluster. You can remove any of them at any time. The first instance does not have special properties. The base configuration is created from this instance, but the instance can be removed from the cluster in the same manner as the other instances.
- Most of the clustering management, configuration, and application deployment is handled through the Oracle Enterprise Manager. If you want to use a command-line tool, you can use the Distributed Configuration Management (DCM) command-line tool, which is documented in [Appendix F, "DCM Command-Line Utility \(dcmctl\)"](#).
- You can remove application server instances from the cluster. The application server instance is stopped when removed from the cluster. When the last application server instance is removed, the cluster still remains. You must delete the cluster itself for it to be removed.

Management Features

Each application server instance contains management features that manage and monitor the application server instance, its components, and how it performs in a cluster. The management features do the following:

- Propagate the cluster-wide configuration for the application server instances and its components
- Manage the application server components by starting, stopping, and restarting these components
- Notice if a component dies and restarts it
- Notify the Oracle HTTP Server if any OC4J instances start or stop

All of these activities are provided by the following management features:

- [Distributed Configuration Management \(DCM\)](#)
- [Oracle Process Management and Notification \(OPMN\)](#)

Distributed Configuration Management (DCM)

Distributed Configuration Management (DCM) manages configuration by propagating the cluster-wide configuration for the application server instances and its components. When you add application server instances to the cluster, it is the

DCM component that automatically replicates the base configuration to all instances in the cluster. When you modify the cluster-wide configuration, DCM propagates the changes to all application server instances in the cluster.

DCM is a management feature in each application server instance. However, it is not a process that exists at all times. DCM is invoked either by Oracle Enterprise Manager or manually by a user through `dcmctl` to do the following:

- Create or remove a cluster
- Add or remove application server instances to or from a cluster
- Synchronize configuration changes across application server instances
- Send application server instance start, restart, and stop requests to OPMN
- Enable automatic re-configuration on system failure

You can also manually execute the DCM command-line tool—`dcmctl`—to perform these duties. However, there are restrictions on how to use `dcmctl`, which are as follows:

- If Oracle Enterprise Manager is up and managing the cluster, you can invoke the DCM command-line tool from any host where a clustered application server instance exists. DCM informs Oracle Enterprise Manager of the requested function. Oracle Enterprise Manager then interfaces with the other DCM management features on the other application server instances in the cluster to complete the cluster-wide function.
- If Oracle Enterprise Manager is not up and managing the cluster, you must start the DCM command-line tool in the foreground on each application server instance in the cluster. Once started in the foreground, the DCM process in each application server instance communicate with each other about configuration changes and deployed applications.

See Also: ["Using dcmctl in a Clustered Environment" on page F-5](#) for directions on how to do the previous functions with the `dcmctl` tool

Oracle Process Management and Notification (OPMN)

Oracle Process Management and Notification (OPMN) manages Oracle HTTP Server and OC4J processes within an application server instance. It channels all events from different components to all components interested in receiving them.

OPMN consists of the following two components:

- Oracle Process Manager
- Oracle Notification Manager

See Also: "Oracle Process Management and Notification" on page 13-6 for detailed information.

Component Instances

The application server is installed with several different types of components. However, to be involved in a cluster, each application server instance can only contain one Oracle HTTP Server and one or more Oracle9iAS Containers for J2EE (OC4J) components. As noted above, Oracle9iAS Web Cache can be installed, but it will not be clustered within this environment. Oracle9iAS Web Cache has its own clustering model.

Note: Other application server components, such as Oracle9iAS Web Cache, can be clustered independently from application server clusters. It is not recommended that a component be part of an independent cluster as well as an application server instance cluster. For information on components that can be clustered independently, see each component administrator's guide.

Oracle HTTP Server

The Oracle HTTP Server is a Web server for the application server instance. It serves client requests. In addition, it forwards OC4J requests to an active OC4J process. Because of this, Oracle HTTP Server is a natural load balancer for OC4J instances. When you have a single application server instance, the Oracle HTTP Server handles the incoming requests for all of the OC4J processes in this sole application server instance. However, in a clustered environment, the Oracle HTTP Server is updated with information about existing OC4J processes by OPMN in all application server instances in the cluster. Thus, the Oracle HTTP Server can do the following:

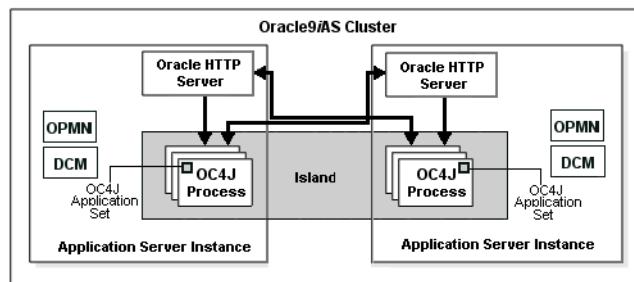
- Forward an incoming stateless request to any OC4J process in the cluster. The priority is to forward the incoming request first to an OC4J process in its own application server instance. If none are available, it forwards the request to any OC4J process in another application server instance in the cluster.
- Forward an incoming stateful request to the particular OC4J process where the conversation originated. If the OC4J process has failed, Oracle HTTP Server

forwards the request to another OC4J process that has the replicated state of that application.

OPMN starts (or restarts) each OC4J process. OPMN notifies each Oracle HTTP Server in the cluster of each OC4J process. Thus, any Oracle HTTP Server can load balance incoming requests among any OC4J process in the cluster.

[Figure 14–6](#) demonstrates how the two Oracle HTTP Servers in the cluster know about both of the OC4J processes. It does not matter that one OC4J process exists in a separate application server instance, which can be installed on a separate host. The OPMN components in each application server instance notifies both Oracle HTTP Servers of the OC4J processes when they were initialized.

Figure 14–6 OC4J Processes in Islands



OC4J Instance

The OC4J instance is the entity to which J2EE applications are deployed and configured. It defines how many OC4J processes exist within the application server and the configuration for these OC4J processes. The OC4J process is what executes the J2EE applications for the OC4J instance.

The OC4J instance has the following features:

- The configuration of the OC4J instance is valid for one or more OC4J executable processes. This way, you can duplicate the configuration for multiple OC4J processes by managing these processes in the OC4J instance construct. When you modify the cluster-wide configuration within the OC4J instance, the modifications are valid for all OC4J processes.

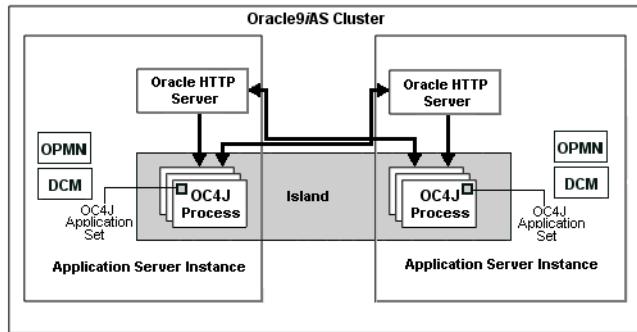
- You can configure each OC4J instance with one or more OC4J processes.
- When you deploy an application to an OC4J instance, the OC4J instance deploys the application to all OC4J processes defined in the OC4J instance. The OC4J instance is also responsible for replicating the state of its applications.
- The number of OC4J processes is specific to each OC4J instance. This must be manually configured for each application server instance in the cluster. The OC4J process configuration provides flexibility to tune according to the specific hardware capabilities of the host. By default, each OC4J instance is instantiated with a single OC4J process.

Within the application sever instance, you can configure multiple OC4J instances, each with its own number of OC4J processes. The advantage of this is for configuration management and application deployment for separate OC4J processes in your cluster.

[Figure 14-7](#) demonstrates the `OC4J_home` default OC4J instance. In the context of a cluster, the OC4J instance configuration is part of the cluster-wide configuration. Thus, the `OC4J_home` instance, configured on the first application instance, is replicated on all other application server instances.

The number of processes in each `OC4J_home` instance is an instance-specific parameter, so you must configure the `OC4J_home` instance separately on each application server instance for the number of OC4J processes that exist on each application server instance. [Figure 14-7](#) shows that the `OC4J_home` instance on application server instance 1 contains two OC4J processes; the `OC4J_home` instance on application server instance 2 contains only one OC4J process. Each OC4J instance defaults to having one OC4J process.

Figure 14–7 OC4J Processes in a Cluster



OC4J Process

The OC4J process is the JVM process that executes J2EE applications. Each OC4J process is contained in an OC4J instance and inherits its configuration from the OC4J instance. All applications deployed to an OC4J instance are deployed to all OC4J processes in the OC4J instance.

You can define one or more OC4J processes within an OC4J instance, so that J2EE requests can be load balanced and have failover capabilities.

The configuration for the number of OC4J processes is instance-specific. Thus, you must configure each OC4J instance in each application server instance with the number of OC4J processes you want to start up for that OC4J instance. The default is one OC4J process.

Each host that you install the application server instances on has different capabilities. To maximize the hardware capabilities, properly configure the number of OC4J processes in each OC4J instance that will use these capabilities. For example, you can configure a single OC4J process on host A and five OC4J processes on host B.

When you define multiple OC4J processes, you enable the following:

- You can serve multiple users with multiple OC4J processes.
- You can provide failover if the state of the application is replicated across multiple OC4J processes.

- Oracle HTTP Server provides load balancing for all OC4J processes in the OC4J instance. The OPMN component notifies each Oracle HTTP Server when a new OC4J process is initiated. Thus, each Oracle HTTP Server in the cluster knows of each OC4J process in the cluster.

Replicating Application State The OC4J processes involved in the cluster can replicate application state to all OC4J processes. Once you configure replication, OC4J handles the propagation of the application state for you.

If one OC4J process fails, then another OC4J process—which has had the application state replicated to it—takes over the application request. When an OC4J process fails during a stateful request, the Oracle HTTP Server forwards the request in the following order:

1. If another OC4J process is active within the same application server instance, Oracle HTTP Server forwards the request to this process.
2. Otherwise, Oracle HTTP Server forwards the state request to an OC4J process in another application server instance in the cluster.

There are two types of failure that you want to protect against: software failure and hardware failure. [Table 14–1](#) lists their differences.

Table 14–1 Failure Types

Failure Types	Avoidance Technique
Software failure occurs when the OC4J process fails.	Multiple OC4J processes in the same OC4J instance. When one OC4J process fails, the Oracle HTTP Server forwards the request to another OC4J process in the same OC4J instance.
Hardware failure occurs when the host goes down.	OC4J processes in the cluster configured on separate hosts. When the first host dies, the OC4J process on another host can take over the request. This requires that you have installed an application server instance on another host, which is a part of the cluster, and the OC4J instance has at least one OC4J process.

Islands

An island is a logical grouping of OC4J processes that allows you to determine which OC4J processes replicate state.

In each OC4J instance, you can have more than one OC4J process. If we consider state replication in a situation where all OC4J processes tried to replicate state, then the CPU load can increase significantly. To avoid a performance degradation, the

OC4J instance enables you to subgroup your OC4J processes. The subgroup is called an island.

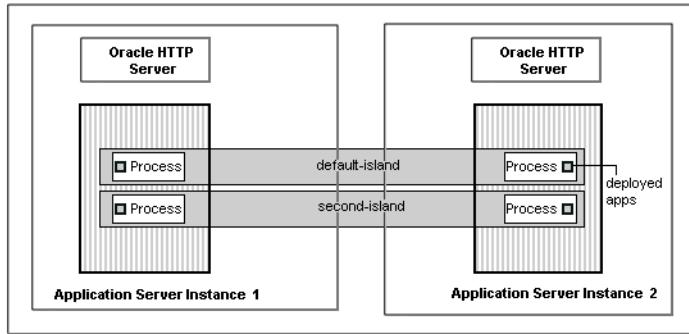
To ensure that the CPU load is partitioned among the processes, the OC4J processes of an OC4J instance can be partitioned into islands. The state for application requests is replicated only to OC4J processes that are grouped within the same island. All applications are still deployed to all OC4J processes in the OC4J instance. The only difference is that the state for these applications is confined to only a subset of these OC4J processes.

The island configuration is instance-specific. The name of the island must be identical in each OC4J instance where you want the island to exist. When you configure the number of OC4J processes on each application server instance, you can also subgroup them into separate islands. The OC4J processes are grouped across application server instances by the name of the island. Thus, the application state is replicated to all OC4J processes within the island of the same name spanning application server instances.

The grouping of OC4J processes for the state replication is different for EJB applications than for Web applications. Web applications replicate state within the island sub-grouping. EJB applications replicate state between all OC4J processes in the OC4J instance and do not use the island sub-grouping.

[Figure 14–8](#) demonstrates OC4J processes in islands within the cluster. Two islands are configured in the `OC4J_home` instance: `default-island` and `second-island`. One OC4J process is configured in each island on each application server instance. The OC4J islands, designated within the shaded area, span application server instances.

Figure 14–8 Island Description



J2EE Applications

J2EE applications are deployed in all cases to the OC4J instance—whether the application server instance is included in a cluster or not. However, when the application is deployed to an OC4J instance that is in a cluster, certain configuration details must be accomplished:

- Multicast host and port—The state of the applications is replicated from one OC4J process to another over a multicast address. In the case of an EJB application, you must also specify a username and password. You can either accept the defaults for the multicast address or configure them through the Oracle Enterprise Manager.
- State replication request—You request state replication for all applications through the Oracle Enterprise Manager.
- XML deployment descriptor elements—Both Web and EJB applications require an additional configuration in their respective XML deployment descriptors.
- Island definition—Web applications use the island subgrouping for its state replication. EJB applications ignore the island subgrouping and use all OC4J processes for its state replication.

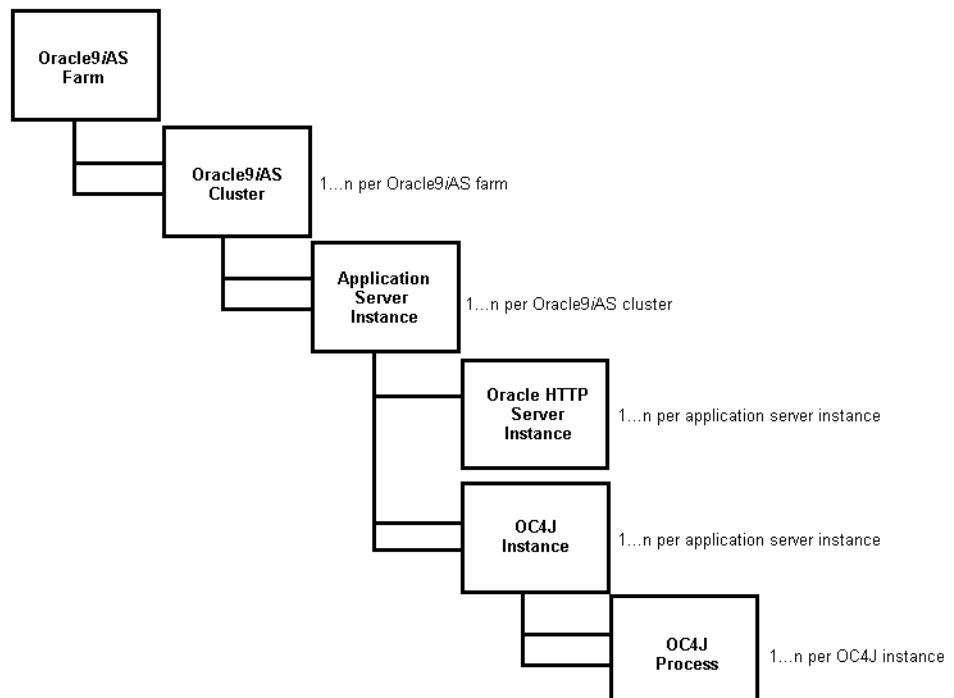
Oracle Enterprise Manager Configuration Tree

Oracle Enterprise Manager uses a hierarchical approach for configuring and managing your cluster.

Figure 14–9 demonstrates the configuration tree for a cluster.

- A cluster contains one or more application server instances.
- Each application server instance contains a single Oracle HTTP Server and one or more OC4J instances.
- Within each OC4J instance, you do the following:
 - Define one or more islands
 - Configure one or more OC4J processes within designated islands
 - Deploy applications

Figure 14–9 Oracle Enterprise Manager Cluster Configuration Tree



Instance-Specific Parameters

The following parameters are not replicated across the cluster.

- Islands and number of OC4J processes—While you want to keep the names of the islands consistent across the application server instances, the definition of the islands and the number of OC4J processes is configured independently. The host on which you install each application server instance has different capabilities. On each host, you can tune the number of OC4J processes to match the host capabilities. Remember that the state is replicated in islands across application boundaries. So the island names must be the same in each OC4J instance.
- Port numbers—The RMI, JMS, and AJP port numbers can be different for each host.
- Command line options—The command line options you use can be different for each host.

Examples

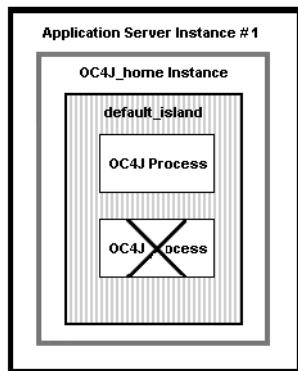
No matter how many application server instances you add within the cluster, the cluster-wide configuration is replicated within the cluster. You control protecting against software and hardware failure with how you configure island and OC4J processes, which are instance-specific parameters.

Software Failure

Suppose you configure more than one OC4J process within your OC4J instance, then if one of these processes fails, another process can take over the work load of the failed process. [Figure 14–10](#) shows application server instance 1, which is involved in the cluster. Within this application server instance, there are two OC4J processes defined in the `default_island` in the `OC4J_home` instance. If the first OC4J process fails, the other can pick up the work load.

Both of these OC4J processes are on the same host; so, if the host goes down, both OC4J processes fail and the client cannot continue processing.

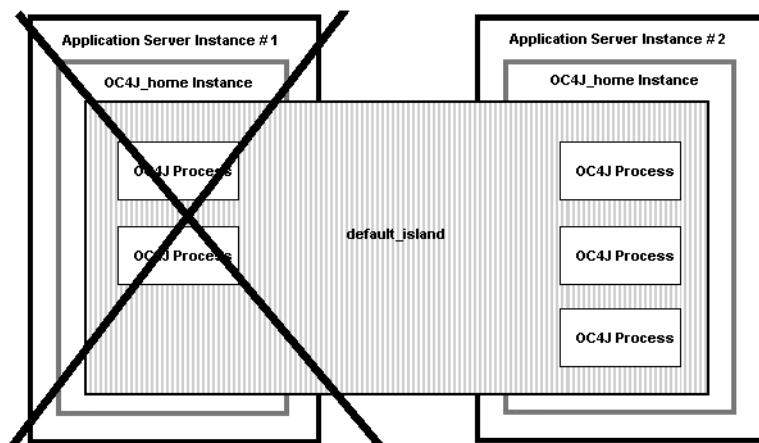
Figure 14–10 Software Failure Demonstration



Hardware Failure

To protect against hardware failure, you must configure OC4J processes in the same OC4J instance across hosts. [Figure 14-11](#) shows OC4J_home instance in application server instance 1 and 2. Within the default-island, two OC4J processes are configured on application server instance 1 and three are configured in application server instance 2. If a client is interacting with one of the OC4J processes in application server 1, which terminates abnormally, the client is redirected automatically to one of the OC4J processes in the default-island in application server 2. Thus, your client is protected against hardware failure.

Figure 14-11 Hardware Failure Demonstration



State Replication

If the client is a stateful application, then the state is replicated only within the same island. In the previous example, there is only a single island, so the state of the application would be preserved.

To enhance your performance, you want to divide up state replication among islands. However, you must also protect for hardware and software failure within these islands.

The optimal method of protecting against software and hardware failure, while maintaining state with the least number of OC4J processes, is to configure at least one OC4J process on more than one host in the same island. For example, if you have application server instance 1 and 2, within the `OC4J_home` instance, you configure one OC4J process in the `default-island` on each application server instance. Thus, you are protected against hardware and software failure and your client maintains state if either failure occurs.

- If one of the OC4J processes fails, then the client request is redirected to the other OC4J process in the island. The state is preserved and the client does not notice any irregularity.
- If application server 1 terminates abnormally, then the client is redirected to the OC4J process in the `default-island` on application server 2. The state is preserved and the client does not notice any irregularity.

As demand increases, you will configure more OC4J processes. To guard against a performance slowdown, separate your OC4J processes into separate islands. For example, if fifteen OC4J processes utilize the hardware efficiently on the two hosts and serve the client demand appropriately, then you could divide these processes into at least two islands. The following shows the fifteen OC4J processes grouped into three islands:

Island Names	Application Server 1	Application Server 2
default-island	two	three
second-island	two	three
third-island	three	two

- The host where application server 1 is installed can handle seven OC4J processes; the host where application server 2 is installed can handle eight OC4J processes.

- Each island contains at least one OC4J process in each island across hosts to protect against software and hardware failure.
- The performance is maximized by dividing up the state replication across three islands.

Cluster Configuration

The following sections describe how to create a cluster and add application server instances to this cluster using Oracle Enterprise Manager:

- [Managing an Oracle9iAS Cluster](#)
- [Managing Application Server Instances in a Cluster](#)
- [OC4J Instance Configuration](#)
- [Configuring Single Sign-On](#)
- [Configuring Instance-Specific Parameters](#)

Note: As an alternative to using Oracle Enterprise Manager, you can create a cluster, add application server instances to the cluster, and manage the cluster using the DCM command-line tool. See [Appendix F, "DCM Command-Line Utility \(dcmctl\)"](#) for information on the DCM command-line tool.

Managing an Oracle9iAS Cluster

From the Oracle9iAS Farm Home Page, you can view a list of all the application server instances that are part of the farm. These application server instances can be clustered.

For more information, see the following topics:

- [Associating an Instance with an Oracle9iAS Infrastructure](#)
- [Creating the Cluster](#)
- [Managing the Cluster](#)

Associating an Instance with an Oracle9iAS Infrastructure

If you have not already done so during installation, you can associate an application server instance with an infrastructure, as follows:

1. Navigate to the Oracle9iAS Instance Home Page.
2. Scroll down to the Administration section and click **Use Infrastructure**.
3. Follow the instructions provided by the Use Infrastructure wizard.

See Also: "[Associating an Instance with an Infrastructure \(Joining a Farm\)](#)" on page 10-4 for more information

Creating the Cluster

Use the Oracle9iAS Farm Home Page to create a new cluster. The Farm Home Page appears when you open the Oracle Enterprise Manager Web site on a host computer that contains an application server instance that is part of a farm.

To create a cluster:

1. Navigate to the Farm Home Page.

[Figure 14–12](#) shows the Farm Home Page with a single application server instance.

Figure 14–12 Oracle9iAS Farm Home Page

The screenshot displays the Oracle9iAS Farm Home Page. At the top left, there is a link to "Standalone Instances". Below it, the title "Farm" is displayed in blue. Under "Farm", the "Clusters" section is shown with the heading "Clusters" and the sub-instruction "All clusters belonging to farm are listed below." A "Create Cluster" button is located in the top right of this section. Below this, there is a table with two columns: "Select Name" and "Status Instances". The "Select Name" column has a radio button next to the value "test". The "Status Instances" column shows a red warning icon and the number "5". In the middle of the page, there is a section titled "Standalone Instances" with the sub-instruction "All instances that belong to farm but are not part of any cluster are listed below." A "Join Cluster" button is located in the top right of this section. Below this, there is another table with three columns: "Select Name", "Status Host", and "Oracle Home". The "Select Name" column has a radio button next to the value "inst1". The "Status Host" column shows a red warning icon and the value "my-sun". The "Oracle Home" column shows the path "/private/oracle". Navigation links "Previous" and "Next" are located above the "Status Instances" table, and "Return to Top" is located above the "Join Cluster" button.

2. Click **Create Cluster**.

Oracle9iAS displays the Create Cluster page. [Figure 14–13](#) shows this page.

Figure 14–13 Create Cluster Page

Create Cluster

Enter the name of the cluster you wish to create.

* Cluster name

3. Enter a name for the cluster and click **Create**.

A confirmation message appears.

4. Click **OK** to return to the Farm Home Page.

The new cluster is listed in the Clusters table.

Managing the Cluster

[Figure 14–14](#) shows the Farm Home Page after a cluster is created.

Figure 14–14 Oracle9iAS Farm Page

Clusters	
All clusters belonging to farm are listed below.	<input type="button" value="Create Cluster"/>
<input checked="" type="radio"/> test	<input type="button" value="Start"/> <input type="button" value="Stop"/> <input type="button" value="Restart"/> <input type="button" value="Delete"/>
<input type="button" value="Previous"/> 1-1 of 1 <input type="button" value="Next"/>	
Status Instances	
<input checked="" type="radio"/> test	5

Standalone Instances		
All instances that belong to farm but are not part of any cluster are listed below.		
<input checked="" type="radio"/> inst1	<input type="button" value="Join Cluster"/>	<input type="button" value="Return to Top"/>
<input type="button" value="Previous"/> 1-1 of 1 <input type="button" value="Next"/>		
Select Name	Status	Host
<input checked="" type="radio"/> inst1	!	my-sun
Oracle Home		
/private/oracle		

[Table 14–2](#) lists the options available on the Farm Home Page.

Table 14–2 Oracle9iAS Farm Page Options

If you want to...	Then...
Start all application server instances in a cluster	Select the radio button next to the cluster and click Start .
Restart all application server instances in a cluster	Select the radio button next to the cluster and click Restart .
Stop all application server instances in a cluster	Select the radio button next to the cluster and click Stop .
Delete a cluster, including any application server instances still included in the cluster.	Select the radio button next to the cluster and click Delete .

Managing Application Server Instances in a Cluster

The following sections discuss how you can manage application server instances in a cluster:

- [Adding an Application Server Instance to a Cluster](#)
- [Removing an Application Server Instance from a Cluster](#)

Adding an Application Server Instance to a Cluster

To add an application server instance to a cluster:

1. Navigate to the Farm Home Page, which is shown in [Figure 14–14](#).
2. Select the radio button of the application server instance in the Standalone Instances section that you want to add to a cluster. In [Figure 14–14](#), the radio button by the `inst1` application server instance is selected.
3. Click **Join Cluster**. [Figure 14–15](#) shows the Join Cluster page.

Figure 14–15 Join Cluster Page

Select Name	Status Instances
<input checked="" type="radio"/> test	5

[Cancel](#) [Join](#)

4. Select the radio button of the cluster that you want the application server instance to join. In [Figure 14–15](#), the `test` cluster is selected.

5. **Click **Join**.**

Oracle9iAS adds the application server instance to the selected cluster and then displays a confirmation page.

6. Click **OK** to return to the Farm Home Page. This moves the application server instance from the standalone instances into the cluster. In doing so, the instance is stopped. You can restart the instance within the context of the cluster.

You will notice that the application server instance disappears from the Standalone Instances section. Also, the number of application server instances displayed for the cluster increases by one. If you display the cluster, you will see that the application server instance was moved into the cluster. Thus, the Standalone Instances section displays only those application server instances that are not a part of any cluster.

Repeat these steps for each additional standalone application server instance you want to add to the cluster.

Removing an Application Server Instance from a Cluster

To remove the application server instance from the cluster, do the following:

1. Select the cluster in which you are interested on the Farm Home Page. This brings you to the cluster page.
2. Select the radio button of the application server instance to remove from the cluster and click **Remove**.

When you add or remove an application server instance to or from a cluster, the application server instance is stopped.

OC4J Instance Configuration

The *Oracle9iAS Containers for J2EE User's Guide* describes how to configure an OC4J Instance. The following sections describe how to configure your OC4J Instance for clustering:

- [Configuring Islands and Processes](#)
- [Configuring Web Application State Replication](#)
- [Configuring EJB Application State Replication](#)

Configuring Islands and Processes

To modify the islands and the number of processes each island contains, do the following:

1. Scroll down to the Administration section of the OC4J Home Page.
2. Select **Server Properties** in the Instance Properties column.
3. Scroll down to the Multiple VM Configuration section. This section defines the islands and the number of OC4J processes that should be started on this application server instance in each island.

Figure 14–16 displays the Multiple VM Configuration section.

Figure 14–16 Island and Process Configuration

Multiple VM Configuration		
Islands		
Island ID	Number of Processes	Related Links
default_island	1	Virtual Machine Metrics
Add Another Row		
Ports		
RMI Ports	3101-3200	
JMS Ports	3201-3300	
AJP Ports	3000-3100	
Command Line Options		
Java Executable		
OC4J Options		
Java Options		

4. Create any islands for this OC4J instance within the cluster by clicking **Add Another Row**. You can supply a name for each island within the Island ID field. You can designate how many OC4J processes should be started within each island by the number configured in the Number of Processes field.

Configuring Web Application State Replication

Configuring state replication for stateful applications is different for Web applications than for EJB applications. To configure state replication for Web applications, do the following:

1. Scroll down to the Administration section of the OC4J Home Page.
2. Select **Replication Properties** in the Instance Properties column.
3. Scroll down to the Web Applications section. [Figure 14–17](#) shows this section.
4. Select the **Replicate session state** checkbox.

Optionally, you can provide the multicast host IP address and port number. If you do not provide the host and port for the multicast address, it defaults to host IP address 230.0.0.1 and port number 9127. The host IP address must be between 224.0.0.2 through 239.255.255.255. Do not use the same multicast address for both HTTP and EJB multicast addresses.

Figure 14–17 Web State Replication Configuration



5. Add the `<distributionable/>` tag to all `web.xml` files in all Web applications. If the Web application is serializable, you must add this tag to the `web.xml` file.

The following shows an example of this tag added to `web.xml`:

```
<web-app>
  <distributionable/>
  <servlet>
    ...
  </servlet>
</web-app>
```

```
</servlet>
</web-app>
```

Configuring EJB Application State Replication

The concepts for understanding how EJB object state is replicated within a cluster are described in the *Oracle9iAS Containers for J2EE Enterprise JavaBeans Developer's Guide and Reference*. To configure EJB replication, you must do the following:

1. Scroll down to the Administration section of the OC4J Home Page.
2. Select **Replication Properties** in the Instance Properties column.
3. Scroll down to the EJB Applications section. [Figure 14–18](#) shows this section.
4. Select the **Replicate State** checkbox.
5. Provide the username and password, which is used to authenticate itself to other hosts in the cluster. If the username and password are different for other hosts in the cluster, they will fail to communicate. You can have multiple username and password combinations within a multicast address. Those with the same username/password combinations will be considered a unique cluster.

Optionally, you can provide the multicast host IP address and port number. If you do not provide the host and port for the multicast address, it defaults to host IP address 230.0.0.1 and port number 9127. The host IP address must be between 224.0.0.2 through 239.255.255.255. Do not use the same multicast address for both HTTP and EJB multicast addresses.

Figure 14–18 EJB State Replication Configuration

EJB Applications

TIP EJB applications replicate state between all OC4J processes in the OC4J instance.

Replicate State

Multicast Host (IP)

Multicast Port

Username

Password

Revert **Apply**

6. Configure the type of EJB replication within the `orion-ejb-jar.xml` file within the JAR file. The type of configuration is dependent on the type of the

bean. See ["EJB Replication Configuration in the Application JAR"](#) on page 14-34 for full details. You can configure these within the `orion-ejb-jar.xml` file before deployment or add this through the Oracle Enterprise Manager screens after deployment. If you add this after deployment, drill down to the JAR file from the application page.

EJB Replication Configuration in the Application JAR Modify the `orion-ejb-jar.xml` file to add the configuration for stateful session beans and entity beans required for state replication. The following sections offer more details:

- [Stateful Session Bean Replication Configuration](#)
- [Entity Bean Replication Configuration](#)

[Stateful Session Bean Replication Configuration](#)

You configure the replication type for the stateful session bean within the bean deployment descriptor. Thus, each bean can use a different type of replication.

- [VM Termination Replication](#)

Set the `replication` attribute of the `<session-deployment>` tag in the `orion-ejb-jar.xml` file to "VMTermination" as follows:

```
<session-deployment replication="VMTermination" .../>
```

- [End of Call Replication](#)

Set the `replication` attribute of the `<session-deployment>` tag in the `orion-ejb-jar.xml` file to "endOfCall" as follows:

```
<session-deployment replication="EndOfCall" .../>
```

- [Stateful Session Context](#)

No static configuration is necessary when using the stateful session context to replicate information across the clustered hosts. To replicate the desired state, set the information that you want replicated and execute the `setAttribute` method within the `StatefulSessionContext` class in the server code. This enables you to designate what information is replicated and when it is replicated. The state indicated in the parameters of this method is replicated to all hosts in the cluster that share the same multicast address, username, and password.

[Entity Bean Replication Configuration](#)

Configure the clustering for the entity bean within its bean deployment descriptor.

Modify the `orion-ejb-jar.xml` file to add the `clustering-schema` attribute to the `<entity-deployment>` tag, as follows:

```
<entity-deployment ... clustering-schema="asynchronous-cache" .../>
```

Configuring Single Sign-On

In order to participate in Single Sign-On functionality, all Oracle HTTP Server instances in a cluster must have an identical Single Sign-On registration.

- Each Oracle HTTP Server is registered with the same Single Sign-On server.
- Each Oracle HTTP Server redirects a success, logout, cancel, or home message to the public network load balancer. In a clustered environment, each Oracle HTTP Server should redirect message URLs to the network load balancer. Since the client cannot access an Oracle HTTP Server directly, the client interacts with the network load balancer.

As with all cluster-wide configuration, the Single Sign-On configuration is propagated among all Oracle HTTP server instances in the cluster. However, the initial configuration is manually configured and propagated. On one of the application server instances, define the configuration with the `ossoreg.jar` tool. Then, DCM propagates the configuration to all other Oracle HTTP Servers in the cluster.

If you do not use a network load balancer, then the Single Sign-on configuration must originate with whatever you use as the incoming load balancer—Web Cache, Oracle HTTP Server, and so on.

See Also: *Oracle9iAS Single Sign-On Administrator's Guide*

To configure a cluster for Single Sign-On, execute the `ossoreg.jar` command against one of the application server instances in the cluster. This tool registers the Single Sign-On server and the redirect URLs with all Oracle HTTP Servers in the cluster.

Run the `ossoreg.jar` command with all of the options as follows, substituting information for the italicized portions of the parameter values.

The values are described fully in [Table 14–3](#).

- Specify the host, port, and SID of the database used by the Single Sign-On server.
- Specify the host and port of the front-end load balancer in each of the redirect URL parameters—`success_url`, `logout_url`, `cancel_url`, and `home_url`.

These should be HTTP or HTTPS URLs depending on the site security policy regarding SSL access to Single Sign-On protected resources.

- Specify the root user of the host that you are executing this tool on in the **-u** option.

```
java -jar ORACLE_HOME/sso/lib/ossoreg.jar
-oracle_home_path ORACLE_HOME
-host sso_database host_name
-port sso_database port_number
-sid sso_database SID
-site_name site name
-success_url http://host.domain:port/osso_login_success
-logout_url http://host.domain:port/osso_logout_success
-cancel_url http://host.domain:port/
-home_url http://host.domain:port/
-admin_id admin_id
-admin_info admin_info
-config_mod_osso TRUE
-u root
-sso_server_version v1.2
```

Table 14–3 SSORegistrar Parameter Values

Parameter	Value
-oracle_home_path <path>	Absolute path to the Oracle home of the application server instance, where you are invoking this tool.
-host <sso_host>	Database host name where Single Sign-On server resides.
-port <sso_port>	Database port where Single Sign-On server resides.
-sid <sso_SID>	Database SID where Single Sign-On server resides.
-site_name <site>	Hostname and port (<i>host :port</i>) of the Web site. You can provide a logical name; however, the hostname and port are helpful to the administrator.
-success_url <URL>	Redirect URL (<i>host.domain:port</i>) for the routine that establishes the partner application session and session cookies. Use HTTP or HTTPS.
-logout_url <URL>	Redirect URL (<i>host.domain:port</i>) for the routine that logs out of the application session.
-cancel_url <URL>	Redirect URL (<i>host.domain:port</i>) to which users are redirected when they cancel authentication.

Table 14–3 SSORegistrar Parameter Values (Cont.)

Parameter	Value
-home_url <URL>	Redirect URL (<code>host.domain:port</code>) for home. This should be a public host.domain and port: HTTP or HTTPS..
-admin_id <name>	(Optional) User name of the <code>mod_osso</code> administrator. This shows up in the Single Sign-On tool as contact information.
-admin_info <text>	(Optional) Additional information about the <code>mod_osso</code> administrator, such as e-mail address. This shows up in the Single Sign-On tool as contact information.

The `SSORegistrar` tool establishes all information necessary to facilitate secure communication between the Oracle HTTP Servers in the cluster and the Single Sign-On server.

When using Single Sign-On with the Oracle HTTP Servers in the cluster, the `KeepAlive` directive must be set to OFF. The reason is because the Oracle HTTP Servers are behind a network load balancer. Thus, if the `KeepAlive` directive is set to ON, then the network load balancer maintains state with the Oracle HTTP Server for the same connection, which results in an HTTP 503 error. Modify the `KeepAlive` directive in the Oracle HTTP Server configuration. This directive is located in the `httpd.conf` file of the Oracle HTTP Server.

Configuring Instance-Specific Parameters

The manageability feature of the cluster causes the configuration to be replicated across all application server instances in the cluster, which is defined as a cluster-wide configuration. However, there are certain parameters where it is necessary to configure them separately on each instance. These parameters are referred to as instance-specific.

The following parameters are instance-specific parameters, which are not replicated across the cluster. You must modify these parameters on each application server instance.

OC4J Instance-Specific Parameters

The following are instance-specific parameters within each OC4J instance:

- islands
- number of OC4J processes

- port numbers
- command-line options

All other parameters are cluster-wide parameters, which are replicated across the cluster.

Figure 14–19 shows the section where you can modify these parameters. These sections are located in the Server Properties off the OC4J Home Page.

Figure 14–19 Non-Replicated Configuration

The screenshot displays the 'Multiple VM Configuration' interface. It includes three main sections: 'Islands', 'Ports', and 'Command Line Options'.
Islands: A table with two columns: 'Island ID' and 'Number of Processes'. One row is present with 'default_island' in the Island ID column and '1' in the Number of Processes column. A button 'Add Another Row' is located below the table.
Ports: Three input fields for RMI Ports (3101-3200), JMS Ports (3201-3300), and AJP Ports (3000-3100).
Command Line Options: Three input fields for Java Executable, OC4J Options, and Java Options.

In the Command Line Options section, you can add debugging options to the OC4J Options line. For more information about debugging in the OC4J process, see <http://otn.oracle.com/tech/java/oc4j>.

Oracle HTTP Server Instance-Specific Parameters

The following are instance-specific parameters in the Oracle HTTP Server.

- ports
- listening addresses
- virtual host information

You can modify the HTTP Server ports and listening addresses on the Server Properties Page, which can be accessed from the HTTP Server Home Page. You can

modify the virtual host information by selecting a virtual host from the Virtual Hosts section on the HTTP Server Home Page.

A

Component Descriptions

This appendix describes the components available in Oracle9*i* Application Server.

See Also: *Oracle9*i* Application Server Concepts Guide* and individual component guides for more information about each component

Oracle9*i*AS Clickstream Intelligence

Oracle9*i*AS Clickstream Intelligence is a business intelligence tool that provides Discoverer-based reports about Web site performance, visitor traffic, Web content effectiveness, and customer loyalty. It collects data about Web site usage by parsing cookies and query strings from Web server logs, and transforms the data into meaningful and concise reports.

Oracle9*i*AS Containers for J2EE (OC4J)

Oracle9*i*AS Containers for J2EE (OC4J) provides a fast, lightweight, highly scalable, easy-to-use, fully certified J2EE environment. OC4J is written entirely in Java and executes on the standard Java Development Kit (JDK) virtual machine. It provides a complete J2EE environment that includes a JSP translator, a Java Servlet engine, an Enterprise JavaBeans (EJB) container, and support for emerging new technologies such as Web Services, SOAP, WSDL, and UDDI.

Oracle9*i*AS Discoverer

Oracle9*i*AS Discoverer is a business intelligence tool that provides a powerful Web-based interface for querying and analyzing data stored in a database. It enables you to define, schedule, and control queries, generate business reports, and share data without having to understand complex database concepts.

Oracle9iAS Forms Services

Oracle9iAS Forms Services is a comprehensive application framework that enables you to deploy Forms applications to the Web. Clients can access Forms applications from Java applets, combining the ease and accessibility of the Web with the complex application support provided by Forms. You can convert existing client/server Forms applications to Web-based Forms applications without changing application code.

Oracle9iAS Interconnect

Oracle9iAS Interconnect (formerly Oracle Applications InterConnect) provides a comprehensive application integration framework that is designed to elevate integration from a coding exercise to a functional modeling exercise, thereby reducing or eliminating the programming effort required for integration. This integration methodology promotes code reuse and reduces the complexity of managing software lifecycles.

Oracle9iAS Personalization

Oracle9iAS Personalization provides real-time personalization for e-business sales channels such as Web stores, application hosting environments, and business call centers. By delivering real-time personalization through an Oracle database, Oracle9iAS Personalization delivers powerful, scalable, real-time personalization for customer areas of interest. This enables e-businesses to deliver tailored, one-to-one customer experiences to Web site users.

Oracle9iAS Portal

Oracle9iAS Portal provides an out-of-the-box portal solution that does not require complex programming and maintenance. It is designed for building, deploying, and maintaining integrated, self-service enterprise portals. Oracle9iAS Portal is a Web-based software environment for building and deploying enterprise portals. It provides a secure, manageable framework for delivering access to distributed software services and information.

Oracle9iAS Reports Services

Oracle9iAS Reports Services is an enterprise reporting service used by information systems departments to produce high-quality production reports. You can dynamically retrieve, format, and distribute database information using a standard Web browser.

Oracle9iAS Single Sign-On

Oracle9iAS Single Sign-On enables users to log in to Oracle9i Application Server features, and gain access to applications for which they are authorized, using a single user name and password. It is fully integrated with Oracle Internet Directory, and supports LDAP-based user and password management through Oracle Internet Directory.

Oracle9iAS Unified Messaging

Oracle9iAS Unified Messaging is a messaging framework that integrates different types of messages from multiple sources into a single location, or "inbox." The messages can be accessed from a single device, regardless of the original message type. Oracle9iAS Unified Messaging frees you to focus on making decisions, rather than on keeping track of multiple telephone numbers, passwords, and access codes.

Oracle9iAS Web Cache

Oracle9iAS Web Cache is a server accelerator caching service that improves the performance, scalability, and availability of frequently visited e-business Web sites. By storing frequently accessed URLs in virtual memory, Oracle9iAS Web Cache eliminates the need to repeatedly process requests for those URLs on the Web server, and it caches both static and dynamically-generated HTTP content from one or more application Web servers.

Oracle9iAS Web Services

Oracle9iAS Web Services supports a set of messaging protocols, programming standards, and network registration and discovery facilities that expose business functions to authorized parties over the Internet from any Web-connected device. Oracle9iAS Web Services makes it easy to develop and deploy services using J2EE and includes Oracle9iAS Containers for J2EE development, deployment, and runtime features.

Oracle9iAS Wireless

Oracle9iAS Wireless is a Portal-based service for delivering information and applications to mobile devices. Using Oracle9iAS Wireless, you can create custom Portal sites that use different kinds of content, including Web pages, Java applications, and XML applications. Oracle9iAS Wireless make diverse information accessible to mobile devices without having to rewrite content for each client device platform.

Oracle Enterprise Manager

Oracle Enterprise Manager offers the following centralized management tools:

- **Oracle Enterprise Manager Web site:** A browser-based graphical tool for managing application server instances, components, and clusters. The Enterprise Manager Web site is installed with every application server instance.
- **Oracle Enterprise Manage Console:** A Java-based graphical tool for managing your entire Oracle platform, including Oracle9i Application Servers, Oracle database servers, and Oracle applications. The Console offers advanced management features, such as an Event system to notify administrators of changes in the environment, and a Job system to automate repetitive tasks. The Console and its related components are installed with the Oracle Management Server as part of the Oracle9iAS Infrastructure.

Oracle HTTP Server

Oracle HTTP Server provides a Web listener and platform for hosting static and dynamic Web pages, and Internet applications using a variety of technologies including Perl, PL/SQL, CGI, and Java. Based on the proven technology of the Apache HTTP Server, Oracle HTTP Server includes significant enhancements that facilitate load balancing, administration, and configuration.

Oracle Internet Directory

Oracle Internet Directory is an LDAP v3-compliant directory that provides fast retrieval and centralized management of user and security information, as well as a central integration point for Oracle9iAS applications and services.

B

Hardware Requirements

This appendix provides the minimum UNIX hardware requirements for Oracle9*i* Application Server.

It contains the following topics:

- [CPU, Disk, and TMP Requirements for Oracle9iAS](#)
- [Memory Requirements for Oracle9iAS](#)

See Also: *Oracle9i Application Server Installation Guide* for Windows hardware requirements

CPU, Disk, and TMP Requirements for Oracle9iAS

[Table B-1](#) lists the CPU, Disk, and TMP requirements for Oracle9iAS.

Table B-1 Oracle9i Application Server CPU, Disk, and TMP Requirements

Hardware Type	Minimum Requirement
CPU ¹	See the <i>Oracle9i Application Server Installation Guide</i>
Disk Space ²	<p>Oracle9i Application Server</p> <ul style="list-style-type: none">■ J2EE and Web Cache: 435 MB■ Portal and Wireless: 1.20 GB■ Business Intelligence and Forms: 1.91 GB■ Unified Messaging: 2.02 GB <p>Oracle9iAS Infrastructure: 1.91 GB</p> <p>Oracle9iAS Developer Kits: 701 MB</p>
TMP or Swap Space	1 GB
/var/tmp Space	<p>Oracle9i Application Server</p> <ul style="list-style-type: none">■ J2EE and Web Cache: 5 MB■ Portal and Wireless: 12 MB■ Business Intelligence and Forms: 95 MB■ Unified Messaging: 95 MB <p>Oracle9iAS Infrastructure: 7 MB</p> <p>Oracle9iAS Developer Kits: 5 MB</p>
Monitor	256 color viewing capability

¹ Oracle recommends a multiple CPU computer.

² The disk space must be available on a single disk. Oracle9i Application Server does not support spanning the installation over multiple disks.

Memory Requirements for Oracle9iAS

Table B–2 lists the memory requirements for Oracle9iAS.

Table B–2 Oracle9i Application Server Memory Requirements

Install Type	Component Configuration	Minimum Memory Requirements
J2EE and Web Cache	<ul style="list-style-type: none"> ■ Oracle HTTP Server ■ Oracle9iAS Containers for J2EE ■ Oracle9iAS Web Cache¹ 	128 MB
All other Oracle9iAS install types	Configure up to three additional Oracle9iAS components	256 MB
All other Oracle9iAS install types	Configure four or more additional Oracle9iAS components	512 MB
Oracle9iAS Infrastructure	<ul style="list-style-type: none"> ■ Oracle9iAS Metadata Repository ■ Oracle9iAS Single Sign-On ■ Oracle Internet Directory ■ Oracle Management Server ■ J2EE and Web Cache (Infrastructure Instance) 	512 MB
Oracle9iAS Developer Kits	<ul style="list-style-type: none"> ■ Oracle XML Developer Kit ■ Oracle9iAS Portal Developer's Kit ■ Oracle9iAS Wireless Developer's Kit ■ Oracle LDAP Developer's Kit ■ J2EE and Web Cache 	128 MB

¹ Memory for Oracle9iAS Web Cache should be based on the following formula: (average HTTP object size) * (maximum number of objects you want to cache). Thus, if you want to cache 1,000,000 objects and the average size of the objects is 3 KB, then set the maximum cache size to at least 3 GB.

C

Metadata Repository Schemas

The metadata repository, which is installed as part of every infrastructure installation, is an Oracle9*i* Enterprise Edition database that is pre-seeded with schemas to support Oracle9*i*AS components and features.

During installation, most metadata repository schemas are assigned randomized passwords and components are configured to access schemas using these passwords. If you want to change the password for any of these schemas, you must do so with the Enterprise Manager Web site. This ensures that the change is registered with the component as well as in the metadata repository.

See Also: "Changing Component Schema Passwords" on page 10-5 for instructions on changing passwords using Enterprise Manager Web site

A few schemas are assigned known passwords so that users can manually access them. You can change their passwords directly in the metadata repository using your preferred method, such as the SQL*Plus ALTER USER command.

Table C-1 lists each metadata repository schema that is used by an Oracle9*i*AS component or feature, provides its tablespace name, and indicates whether the password should be changed using the Enterprise Manager Web site, or directly in the metadata repository using your preferred method.

Table C-1 Metadata Repository Schemas

Schema	Tablespace	Method for Changing Password
SCOTT	SYSTEM	Directly in metadata repository using preferred method (eg: SQL*Plus) <i>Note: It is not recommended that you change the SCOTT password since some demos may not work as a result.</i>
SYS	SYSTEM	Directly in metadata repository using preferred method (eg: SQL*Plus)
SYSTEM	SYSTEM	Directly in metadata repository using preferred method (eg: SQL*Plus)
Oracle9iAS Clickstream Intelligence		
CLKANA	CLK	EM Web site
CLKRT	CLK	EM Web site
Distributed Configuration Management (DCM)		
DCM	DCM	EM Web site
Oracle9iAS Discoverer		
DISCOVERER5	DISCO_PTMS_META	EM Web site
Oracle9iAS InterConnect (OAI)		
OAIHUB902	USERS	EM Web site
Oracle Certificate Manager (OCM)		
OCM_DB_ADMIN	OCM	EM Web site

Table C-1 Metadata Repository Schemas (Cont.)

Schema	Tablespace	Method for Changing Password
Oracle Enterprise Manager		
OEM_REPOSITORY	OEM_REPOSITORY	Directly in metadata repository using preferred method (eg: SQL*Plus)
Oracle Internet Directory		
ODS	OLTS_DEFAULT	Directly in metadata repository using preferred method (eg: SQL*Plus)
ODSCOMMON	OLTS_DEFAULT	Directly in metadata repository using preferred method (eg: SQL*Plus)
Oracle9iAS Portal		
PORTAL	PORTAL	EM Web site
PORTAL_APP	PORTAL	EM Web site
PORTAL_DEMO	PORTAL	EM Web site
PORTAL_PUBLIC	PORTAL	EM Web site
Oracle9iAS Single Sign-On		
ORASSO	USERS	EM Web site
ORASSO_DS	USERS	EM Web site
ORASSO_PA	USERS	EM Web site
ORASSO_PS	USERS	EM Web site
ORASSO_PUBLIC	USERS	EM Web site
Oracle9iAS Syndication Server		
DSGATEWAY	DS_GATEWAY	EM Web site
Oracle9iAS Web Services		
DSSYS	DS_REGISTRY	EM Web site
UDDISYS	UDDISYS_TS	EM Web site

Table C–1 Metadata Repository Schemas (Cont.)

Schema	Tablespace	Method for Changing Password
Oracle9iAS Wireless		
WIRELESS	USERS	EM Web site
Oracle Workflow		
BLEWIS	USERS	EM Web site
CDOUGLAS	USERS	EM Web site
KWALKER	USERS	EM Web site
OWF_MGR	USERS	EM Web site
SYSADMIN	USERS	EM Web site
WFADMIN	USERS	EM Web site

D

Default Port Numbers and Port Ranges

Oracle9iAS automatically assigns port numbers to a component when the component is configured. It chooses the port number from a pre-allocated set of default port numbers and port ranges.

Oracle9iAS uses the following method to assign port numbers:

1. Determine if the default port number is already in use by an Oracle9iAS or non-Oracle9iAS process.
2. If the default port number is not in use, assign it to the component.
3. If the default port number is already in use, attempt to assign a number from the port range, starting with the lowest number, until a free port number is found.

See Also: ["Task 2: Check for Port Conflicts"](#) on page 4-3 for more information about port numbers.

This appendix contains the following tables:

- [Oracle9iAS Port Usage \(Sorted by Component\)](#)
- [Oracle9iAS Port Usage \(Sorted by Port Number\)](#)

Oracle9iAS Port Usage (Sorted by Component)

[Table D-1](#) lists Oracle9iAS ports by components names.

Table D-1 Oracle9iAS Port Usage (Sorted by Component)

Component	Default Port Number	Port Number Range
Oracle9iAS Clickstream Intelligence		
Oracle9iAS Clickstream Collector Server	6675	Fixed
Oracle9iAS Clickstream Execution Engine	6676	Fixed
Oracle9iAS Clickstream Intelligence Viewer	Same as Oracle HTTP Server	Same as Oracle HTTP Server
Oracle9iAS Clickstream Runtime Administrator	Same as Oracle HTTP Server	Same as Oracle HTTP Server
Oracle9iAS Containers for J2EE (OC4J)		
OC4J AJP	3001	3001-3100
OC4J RMI	3101	3101-3200
OC4J JMS	3201	3201-3300
OC4J HTTP Listener	3301	3301-3400
Oracle9iAS Discoverer		
Oracle9iAS Discoverer	Same as Oracle HTTP Server	Same as Oracle HTTP Server
IIOP	16001	16001-16020
Oracle9iAS Forms Services		
Oracle9iAS Forms Services	Same as Oracle HTTP Server	Same as Oracle HTTP Server

Table D-1 Oracle9iAS Port Usage (Sorted by Component) (Cont.)

Component	Default Port Number	Port Number Range
Oracle HTTP Server		
Oracle HTTP Server- non-SSL	7777	7777-7877
Oracle HTTP Server- SSL	4443	4443-4543
Oracle HTTP Server Listen- non-SSL	7777	7777-7877
Oracle HTTP Server- non-SSL- Oracle9iAS Web Cache is installed and configured	7778	7777-7877
Oracle HTTP Server- SSL	4443	4443-4543
Oracle HTTP Server- SSL- Oracle9iAS Web Cache is installed and configured	4444	4443-4543
Oracle HTTP Server JServ Servlet Engine	8007	8007-8107
Oracle HTTP Server- Oracle Notification Service Requested Port	6003	6003-6099
Oracle HTTP Server- Oracle Notification Service Local Port	6100	6100-6199
Oracle HTTP Server- Oracle Notification Service Remote Port	6200	6200-6299
Oracle HTTP Server- Java Object Cache	7000	7000-7010
Oracle9iAS Clickstream Intelligence Collector Agent	6666	6666-6674
Oracle9iAS Portal		
Oracle9iAS Portal	Same as Oracle HTTP Server	Same as Oracle HTTP Server
Oracle9iAS Single Sign-On	5000-5099	5000-5099
Oracle9iAS Reports Services		
Oracle9iAS Reports Services	3000	3000-3010
SQL*Net- For 6i Backward Compatibility Only	1950	1950-1960
Visigenics CORBA- Reports 9i	14000	14000-14010

Table D–1 Oracle9iAS Port Usage (Sorted by Component) (Cont.)

Component	Default Port Number	Port Number Range
Oracle9iAS Unified Messaging		
Oracle9iAS Unified Messaging	5100-5200	5100-5200
IMAP4	143	Fixed
IMAP4- SSL	993	Fixed
POP3	110	Fixed
POP3- SSL	995	Fixed
SMTP	25	Fixed
NNTP	119	Fixed
NNTP- SSL	563	Fixed
Oracle9iAS Web Cache		
Oracle9iAS Web Cache HTTP Listen- non-SSL	7777	7777-7877
Oracle9iAS Web Cache HTTP Listen- SSL	4443	4443-4543
Oracle9iAS Web Cache Administration	4000	4000-4030
Oracle9iAS Wireless		
Oracle9iAS Wireless	Same as Oracle HTTP Server	Same as Oracle HTTP Server
Oracle9iAS Wireless Integration Server	5555	9000-9100
Oracle Enterprise Manager		
Oracle Enterprise Manager Application Server Administration Service	1810, 1811	1812-1820
Oracle Enterprise Manager Intelligent Agent	1748, 1754, 1808, 1809	Fixed
Oracle Management Server	7771, 7772, 7773	7771-7773

Table D-1 Oracle9iAS Port Usage (Sorted by Component) (Cont.)

Component	Default Port Number	Port Number Range
Oracle Internet Directory		
Oracle Internet Directory- non-SSL	389	4031-4040
Oracle Internet Directory- SSL	636	4031-4040
Oracle Internet File System		
Oracle Internet File System Domain Controller and Nodes	53140	Fixed
Oracle Internet File System Domain Controller	53140	Fixed
Main Oracle Internet File System Node	53141, 53142	53141-53142
Oracle Internet File System HTTP Node	53143, 53144	53143-53144
LDAP- non-SSL	Same as Oracle Internet Directory	Same as Oracle Internet Directory
LDAP- SSL	Same as Oracle Internet Directory	Same as Oracle Internet Directory
AFP	548	Fixed
FTP	21	2100-2100
NFS	2049	4048-4049 for Mount Server
IMAP	Same as Oracle Internet Directory	Same as Oracle Internet Directory
IMAP4- SSL	Same as Oracle Internet Directory	Same as Oracle Internet Directory
Servlet	13138	13138-13140
SMTP	Same as Oracle Internet Directory	Same as Oracle Internet Directory
SMB	139	Fixed
CUP	4180	4180-4200
WCP	Same as Oracle Internet Directory	Same as Oracle Internet Directory
Oracle Workflow		
TNS	1521	Fixed

Oracle9iAS Port Usage (Sorted by Port Number)

[Table D-2](#) lists Oracle9iAS ports in ascending order.

Table D-2 Oracle9iAS Port Usage (Sorted By Port Number)

Port Number	Component
21	Oracle Internet File System - FTP
25	Oracle9iAS Unified Messaging - SMTP
110	Oracle9iAS Unified Messaging - POP3
119	Oracle9iAS Unified Messaging - NNTP
139	Oracle Internet File System - SMB
143	Oracle9iAS Unified Messaging IMAP4
389	Oracle Internet Directory- non-SSL
548	Oracle Internet File System - AFP
563	Oracle9iAS Unified Messaging NNTP- SSL
636	Oracle Internet Directory- SSL
993	Oracle9iAS Unified Messaging - IMAP4-SSL
995	Oracle9iAS Unified Messaging POP3- SSL
1521	Oracle Workflow - TNS
1748	Oracle Enterprise Manager Intelligent Agent
1754	Oracle Enterprise Manager Intelligent Agent
1808	Oracle Enterprise Manager Intelligent Agent
1809	Oracle Enterprise Manager Intelligent Agent
1810 - 1820	Oracle Enterprise Manager Application Server Service
1950 - 1960	Oracle9iAS Reports Services SQL Net
2049	Oracle Internet File System - NFS
2070	Oracle9iAS Syndication Server; To access OSS
3000 - 3010	Oracle9iAS Reports Services
3001 - 3100	Oracle9iAS Containers for J2EE - AJP
3101 - 3200	Oracle9iAS Containers for J2EE - RMI

Table D-2 Oracle9iAS Port Usage (Sorted By Port Number) (Cont.)

Port Number	Component
3201 - 3300	Oracle9iAS Containers for J2EE - JMS
3301 - 3400	Oracle9iAS Containers for J2EE HTTP Listener
4000 - 4030	Oracle9iAS Web Cache Administration Port
4001	Oracle9iAS Web Cache Invalidation Port
4002	Oracle9iAS Web Cache Statistics
4031 - 4040	Oracle Internet Directory- SSL and non-SSL
4048 - 4049	Oracle Internet File System - NFS (for Mount Server)
4180 - 4200	Oracle Internet File System - CUP
4443 - 4543	Oracle HTTP Server- SSL, Oracle HTTP Server Listen- SSL, Oracle9iAS Web Cache Listen- SSL
5000 - 5099	Oracle9iAS Single Sign-On
5100 - 5200	Oracle9iAS Unified Messaging
5555	Oracle9iAS Wireless Web Integration Server
6003	Oracle HTTP Server- Oracle Notification Service Request Port
6100	Oracle HTTP Server- Oracle Notification Service Local Port
6200	Oracle HTTP Server- Oracle Notification Service Remote Port
6666	Oracle9iAS Clickstream Collector Agent
6675	Oracle9iAS Clickstream Collector Server
6676	Oracle9iAS Clickstream Execution Engine
6003 - 6099	Oracle HTTP Server - Oracle Notification Service Requested Port
6100 - 6199	Oracle HTTP Server - Oracle Notification Service Local Port
6200 - 6299	Oracle HTTP Server - Oracle Notification Service Remote Port
6666 - 6674	Oracle9iAS Clickstream Intelligence Collector Agent
7000 - 7010	Oracle HTTP Server Java Object Cache
7771 - 7773	Oracle Management Server
7777 - 7877	Oracle HTTP Server- non-SSL, Oracle HTTP Server Listen- non-SSL, Oracle9iAS Web Cache Listen- non-SSL
9000 - 9100	Oracle9iAS Wireless Web Integration Server

Table D–2 Oracle9iAS Port Usage (Sorted By Port Number) (Cont.)

Port Number	Component
8007 - 8107	Oracle HTTP Server JServ Servlet Engine
13138 - 13140	Oracle Internet File System Servlet
14000 - 14010	Oracle9iAS Reports Services Visigenics - CORBA
16001 - 16020	IOP
53140	Oracle Internet File System Domain Controller and Nodes
53141 - 53142	Oracle Internet File System Main Node
53143 - 53144	Oracle Internet File System HTTP Node

E

Configuration Files and Log Files

This appendix summarizes important configuration files and log files for Oracle9*i* Application Server.

- [Configuration Files](#)
- [Log Files](#)

Configuration Files

[Table E-1](#) presents a list of important Oracle9*i* Application Server configuration files. Directory paths are listed in UNIX format; they can be converted to the corresponding Windows paths by inverting the backslashes, except where noted.

Warning: If you edit Oracle HTTP Server, OC4J, or OPMN configuration files manually instead of using the Oracle Enterprise Manager Web site, you must use the DCM command-line utility (dcmctl) to notify the DCM Repository of the changes as follows:

```
dcmctl updateConfig
```

The applicable components are noted with a footnote in [Table E-1](#).

Table E–1 Important Oracle9iAS Configuration Files

Configuration File Name	Configuration File Location²	Corresponding File in Oracle9i Application Server v1.0.2.x
Oracle HTTP Server¹		
httpd.conf oracle_apache.conf access.conf magic mime.types mod_oc4j.conf srm.conf	<i>ORACLE_HOME</i> /Apache/Apache/conf	Same
JServ¹		
jserv.conf jserv.properties zone.properties	(UNIX) <i>ORACLE_HOME</i> /Apache/Jserv/etc (Windows) <i>ORACLE_HOME</i> \Apache\Jserv\conf	Same
mod_oradav¹		
moddav.conf	<i>ORACLE_HOME</i> /Apache/oradav/conf	None
mod_plsql¹		
dads.conf	<i>ORACLE_HOME</i> /Apache/modplsql/conf	<i>ORACLE_HOME</i> /Apache/modplsql/cfg/wdbsvr.app
cache.conf	<i>ORACLE_HOME</i> /Apache/mod_plsql/conf	<i>ORACLE_HOME</i> /Apache/mod_plsql/cfg/cache.cfg
Oracle9iAS Web Cache		
internal.xml internal_admin.xml webcache.xml	<i>ORACLE_HOME</i> /webcache	Same

Table E–1 Important Oracle9iAS Configuration Files (Cont.)

Configuration File Name	Configuration File Location²	Corresponding File in Oracle9i Application Server v1.0.2.x
Oracle9iAS Portal		
targets.xml	<i>ORACLE_HOME</i> /sysman/emd	None
oracle_portal.xml oracle_sso_server.xml	<i>ORACLE_HOME</i> /sysman/admin/metadata	None
AsynchNotification.properties AsynchRequest.properties Cookies.properties EncodingSets.properties Ftp.properties LocaleToCharsetMap.properties LocationMark.properties LocationMark_en_US.properties MimeTypes.properties Notification.properties Provisioning.properties ProxyFirewall.properties System.properties System.properties.teml UserAgents.properties persistent.properties	<i>ORACLE_HOME</i> /wireless/server/classes/oracle/panama/core/admin	Same
WebBeanProperties.properties	<i>ORACLE_HOME</i> /wireless/server/classes/oracle/panama/core/tools/webbean	None
XMLXSL.properties	<i>ORACLE_HOME</i> /wireless/server/classes/oracle/panama/core/xml	Same
YPCategories.xml	<i>ORACLE_HOME</i> /wireless/server/classes/oracle/panama/spatial/yp	Same

Table E–1 Important Oracle9iAS Configuration Files (Cont.)

Configuration File Name	Configuration File Location ²	Corresponding File in Oracle9i Application Server v1.0.2.x
Oracle9iAS Discoverer		
configuration.xml	ORACLE_HOME/j2ee/OC4J_BI_Forms/applications/discoverer/web/WEB-INF/lib	None
viewer_config.xml	ORACLE_HOME/j2ee/OC4J_BI_Forms/applications/discoverer/web/viewer_files	None
plus_config.xml	ORACLE_HOME/j2ee/OC4J_BI_Forms/applications/discoverer/web/plus_files	None
portal_config.xml	ORACLE_HOME/j2ee/OC4J_BI_Forms/applications/discoverer/web/portal	None
pref.txt	ORACLE_HOME/discoverer902/util	ORACLE_HOME/discwb4/util/pref.txt
.reg_key.dc	ORACLE_HOME/discoverer902/bin/.reg	ORACLE_HOME/discwb4/bin/.reg/.reg_key.dc
Oracle9iAS Reports Services		
reports_server_name.conf cgicmd.dat jdbcpds.conf proxyinfo.xml rwbuilder.conf rwserver.template rw servlet.properties textpds.conf xmlpds.conf	ORACLE_HOME/reports/conf	None
Oracle9iAS Clickstream Intelligence		
agent.xml click-apache.conf click-app.properties collector-agent.properties	ORACLE_HOME/click/conf	None
Oracle9iAS Forms Services		
default.env	ORACLE_HOME/forms90/server	None
formsweb.cfg	ORACLE_HOME/forms90/server	Same

Table E–1 Important Oracle9iAS Configuration Files (Cont.)

Configuration File Name	Configuration File Location²	Corresponding File in Oracle9i Application Server v1.0.2.x
Oracle9iAS Personalization		
web.xml	ORACLE_HOME/j2ee/OC4J_BI_Forms/applications/OP/op/WEB-INF	None
Oracle9iAS Unified Messaging		
oc4j.properties	ORACLE_HOME/j2ee/OC4J_UM/config	None
plsql.conf	ORACLE_HOME/Apache/modplsql/conf	None
mod_osso.conf	ORACLE_HOME/Apache/Apache/conf	None
toolkit.properties	ORACLE_HOME/um/client/config	None
DBToLDAPMigrationTool.properties	ORACLE_HOME/oes/calserver/applications/scheduler/customizations/migrate/addrbookmig	None
Oracle Process Manager and Notification		
ons.conf opmn.xml	ORACLE_HOME/opmn/conf	None
Oracle9iAS Single Sign-On Server		
dads.conf	ORACLE_HOME/Apache/modplsql/conf	ORACLE_HOME/Apache/modplsql/cfg/wdbsvr.app
plsql.conf	ORACLE_HOME/Apache/modplsql/conf	ORACLE_HOME/Apache/modplsql/cfg/plsql.conf

1 See warning on page E–2 regarding manually editing Oracle HTTP Server configuration files.

2 Configuration file locations are for UNIX systems. Invert the backslash for Windows locations, except where noted.

Log Files

Table E-2 presents a list of important Oracle9i Application Server log files. Directory paths are listed in UNIX format; they can be converted to the corresponding Windows paths by inverting the backslashes.

Table E-2 Important Oracle9iAS Log Files

Log File Location	Description
<i>ORACLE_HOME/admin/sid/*dump/*</i>	Database creation and related logs.
<i>ORACLE_HOME/assistants/opca/install.log</i>	Portal Configuration.
<i>ORACLE_HOME/webcache/logs/*</i>	Webcache logs.
<i>ORACLE_HOME/dcm/logs/*</i>	DCM logs.
<i>ORACLE_HOME/ldap/log/*</i>	OID log files.
<i>ORACLE_HOME/opmn/logs/*</i>	Apache logs.
<i>ORACLE_HOME/sysman/log/*</i>	EM logs.
<i>ORACLE_HOME/j2ee/OC4J_instance/log/*/*</i>	J2EE logs for each OC4J Instance.
<i>ORACLE_HOME/config/schemaload.log</i>	Contains all schema names associated with metadata repository.
<i>ORACLE_HOME/config/useinfratool.log</i>	Use Infrastructure Tool logs.

DCM Command-Line Utility (dcmctl)

The Distributed Configuration Management (DCM) utility, `dcmctl`, provides a command-line alternative to using Oracle Enterprise Manager for some management tasks. The `dcmctl` tool uses the same distributed architecture and synchronization features as Enterprise Manager Web site, thereby providing identical functionality in a format that is ideal for scripting and automation.

The following sections describe the tasks you can perform using `dcmctl`:

- [Overview](#)
- [Starting and Stopping](#)
- [Managing Application Server Instances](#)
- [Managing Components](#)
- [Managing Clusters](#)
- [Deploying Applications](#)
- [Using the dcmctl Shell](#)
- [Executing dcmctl from a Command File](#)

Overview

The `dcmctl` utility is located in:

(UNIX) `ORACLE_HOME/dcm/bin/dcmctl`
(Windows) `ORACLE_HOME\dcm\bin\dcmctl`

Note: The only type of application server instance that you can manage with `dcmctl` is a J2EE and Web Cache instance type with Oracle HTTP Server (OHS) and Oracle9iAS Containers for J2EE (OC4J) configured. If Web Cache is configured, it will be ignored by `dcmctl`.

In order to run `dcmctl` you must log in to your operating system as the user that installed Oracle9i Application Server. You can run `dcmctl` from your operating system prompt using the following syntax:

```
dcmctl command [options]
```

Table F-1 displays `dcmctl` help and error information commands.

Table F-1 Help and Error Commands

Command	Description
<code>help</code>	View usage information.
<code>getError [err_number err_name]</code>	View a description of the most recent error that occurred if no parameter is given. If you provide an error number or name, the description for that error is displayed. An example of a valid number is 906007; an example of a valid name is ADMN-906007.
<code>getReturnStatus</code>	Print the current status of the last command executed. The last command must be a command that performs an operation, not a command that returns state. If the last command has a failed or unknown state, the <code>-verbose</code> option will provide more information.

The following sections describe overall information on how to use `dcmctl`:

- [About dcmctl Commands and Options](#)
- [Using dcmctl in a Clustered Environment](#)
- [Passing Parameters to the JVM](#)

About dcmctl Commands and Options

The `dcmctl` utility supports many commands, which are described in the subsequent sections of this appendix. Commands are a single word and are not case-sensitive. Each `dcmctl` command supports zero or more options.

Options take the following form:

`-option [argument]`

Option names have a long and short form, and are not case-sensitive. There are two types of `dcmctl` options: target and universal.

Target Options

[Table F-2](#) lists the `dcmctl` target options that define the target on which to apply the given command. Subsequent sections of this appendix describe which target options can be used with each command. On hosts with multiple application server instances, `dcmctl` determines the target instance as follows:

- The target is all instances in the designated cluster with the `-cluster` or `-cl` option.
- The target is the instance supplied with the `-instance` or `-i` option.
- If a cluster or instance is not supplied, then the target is the instance associated with the `-oraclehome` universal option.

- If the cluster, instance, or `-oraclehome` is not supplied, use the instance associated with the Oracle home directory in which the `dcmctl` executable resides.

Table F-2 *dcmctl Target Options*

Option	Description
<code>-application app_name</code> <code>-a app_name</code>	Apply the command to the named application
<code>-cluster cluster_name</code> <code>-cl cluster_name</code>	Apply the command to the named application server cluster
<code>-component comp_name</code> <code>-co comp_name</code>	Apply the command to the named component
<code>-componentType type</code> <code>-ct type</code>	Apply the command to components of the named component type. Component type can be <code>ohs</code> , <code>oc4j</code> , <code>opmn</code> , or <code>jazn</code> .
<code>-instance inst_name</code> <code>-i inst_name</code>	Apply the command to the named application server instance

Universal Options

[Table F-3](#) lists the `dcmctl` universal options that define command behavior and can be used with all commands.

Table F-3 *dcmctl Universal Options*

Option	Description
<code>-debug</code> <code>-d</code>	Print the stack trace if an exception occurs when executing the command
<code>-logdir directory</code> <code>-l directory</code>	Save the DCM error log file <code>log.xml</code> in the named directory. The directory can be a full pathname or a pathname relative to the current directory. The default directory is <code>ORACLE_HOME/dcm/logs</code> .
<code>-oraclehome directory</code> <code>-o directory</code>	Set the Oracle home to the named directory. The default is the Oracle home where the <code>dcmctl</code> command resides.
<code>-timeout num_seconds</code> <code>-t num_seconds</code>	Set the maximum number of seconds to allow for a command to complete. The default is 45 seconds.

Table F-3 dcmctl Universal Options (Cont.)

Option	Description
-verbose	Print the long version of state and error messages
-v	

Using dcmctl in a Clustered Environment

In order to use `dcmctl` in a clustered environment, you must have a DCM daemon associated with every instance in the cluster. You can do this in one of the following ways:

- Start the Oracle Enterprise Manager Web site on each host that contains an application server instance in your cluster. On each host, log in as the user that installed Oracle9i Application Server and enter the following command in the Oracle home of the primary installation (the primary installation is the first application server or infrastructure installed on the system):

```
(UNIX) ORACLE_HOME/bin/emctl start
(Windows) ORACLE_HOME\bin\emctl start
```

See Also: "Starting and Stopping the Oracle Enterprise Manager Web Site" on page 3-9 for more information on starting the Oracle Enterprise Manager Web site.

- On UNIX, you can start the `dcmctl` shell in each application server instance in the cluster. On each host that contains instances in the cluster, log in as the user that installed Oracle9i Application Server and execute the following command in the Oracle home directory for each instance in the cluster:

```
ORACLE_HOME/dcm/bin/dcmctl shell
```

To stop the process, use the following command:

```
dcmctl> exit
```

Note that the `dcmctl` shell must stay running in the foreground; do not put it in the background.

Passing Parameters to the JVM

You can pass parameters directly to the JVM when executing `dcmctl` through the `ORACLE_DCM_JVM_ARGS` environment variable.

For example, to set up a proxy:

```
ORACLE_DCM_JVM_ARGS="-DhttpProxy.host=yourproxyhost.com  
-DhttpProxy.port=yourproxypport"
```

Starting and Stopping

You can use `dcmctl` to start, stop, restart, and retrieve the status of application server instances, components, and clusters.

Table F-4 lists the administration commands and their options for starting, stopping, restarting, and retrieving the status of instances, clusters, or components within the instance or cluster.

Table F-4 Starting and Stopping Commands and Their Options

Command	Description
<code>start [[-cl <i>cluster_name</i>] [-i <i>instance_name</i>] [-co <i>component_name</i>] [-ct <i>type</i>]]</code>	Start the processes indicated. The default is to start the local application server instance only. Refer to Table F-2 for information on the scope parameters. Note that you can choose to start all application server instances in the cluster (-cl), the local application server instance (default), a remote application server instance (-i), a single component within the local instance (-co), or a component type (ohs, oc4j, or opmn) within an instance (-ct). For all options except the -co and -ct options, OPMN and DCM are started if not already executing.
<code>stop [[-cl <i>cluster_name</i>] [-i <i>instance_name</i>] [-co <i>component_name</i>] [-ct <i>type</i>]]</code>	Stop the processes indicated. See the start command for further discussion. This does not stop OPMN and DCM.
<code>restart [[-cl <i>cluster_name</i>] [-i <i>instance_name</i>] [-co <i>component_name</i>] [-ct <i>type</i>]]</code>	Restart the processes indicated. See the start command for further discussion. This will leave OPMN and DCM running.
<code>shutdown</code>	Stop the local application server instance, including its components, OPMN, and DCM. This command is appropriate to run before a system shutdown.
<code>getstate [[-cl <i>cluster_name</i>] [-i <i>instance_name</i>] [-co <i>component_name</i>]]</code>	Return the current status of the processes indicated. This command returns a status of "up" or "down" for the indicated process.

Managing Application Server Instances

Table F-5 describes commands that you can use to display information about application server instances and destroy and resynchronize instances.

Table F-5 Application Server Instance Commands and Their Options

Command	Description
listInstances	Return the names of all instances that belong to the farm as the target instance and are not part of a cluster. If the target instance does not belong to a farm, return only the name of the target instance.
whichInstance	Return the name of the target instance.
destroyInstance -i <i>instance_name</i>	Remove all information related to the instance from the DCM repository. This command can be used if an instance was not deinstalled properly using Oracle Universal Installer. Note that the instance name must be supplied for this command.
listComponents [[-i <i>instance_name</i>] [-cl <i>cluster_name</i>]]	Return a list of component instance names in the application server instance.
resyncInstance [-force] [-i <i>instance_name</i>]	Resynchronize the local configuration information for an instance with what is in the DCM repository. This command can be used if an instance was not able to be updated due to a system failure, and the instance state is not in sync with the DCM repository. The <i>-force</i> option causes an instance to resynchronize with information from the DCM repository regardless of whether the instance state indicates that it requires resynchronization.

Managing Components

Table F-6 describes commands that you can use to manage Oracle HTTP Server and OC4J instances that reside within a J2EE and Web Cache instance type.

Table F-6 Component Commands and Their Options

Command	Description
listComponentTypes	Return a list of supported component types. The current supported component types are ohs, oc4j, opmn, and jazn.
getComponentType -co <i>component_name</i> [-i <i>instance_name</i>]	Return the type of the component instance.
createComponent -ct <i>type</i> -co <i>component_name</i>	Create a new component instance of the specified type with the specified name. Only the oc4j type is allowed.
removeComponent -co <i>component_name</i>	Remove the specified component from the local instance (and from the cluster if applicable). Only oc4j component instances can be removed.
updateConfig [-ct <i>type</i> [, <i>type</i>]]	Update the DCM repository with configuration changes made by manually editing the component configuration files. The componentType indicates the component type that has been edited (ohs, oc4j, opmn, or jazn). The default is both component types.

Managing Clusters

[Table F-7](#) describes commands that you can use to manage application server clusters.

Table F-7 Cluster Commands and Their Options

Command	Description
<code>createCluster -cl cluster_name</code>	Create a cluster with the indicated name in the farm.
<code>removeCluster -cl cluster_name</code>	Remove the cluster and destroy all information about the cluster in the DCM repository. A cluster must contain zero instances when it is removed.
<code>listInstances [-cl cluster_name]</code>	Return a list of application server instances in the cluster. If no cluster is supplied, list instances that are not in a cluster.
<code>listClusters</code>	Return a list of cluster names in the farm that is associated with this host.
<code>whichCluster [-i instance_name]</code>	Return the name of the cluster that contains the supplied instance.
<code>isClusterable [-i instance_name]</code>	Determine if an application server instance is eligible for clustering. Only J2EE and Web Cache instance types with HTTP Server and OC4J configured are eligible. Note that the <code>-verbose</code> option will describe why an instance is not eligible for clustering.
<code>isCompatible -cl cluster_name [-i instance_name]</code>	Determine if an application server instance is compatible with a cluster, and therefore eligible to join the cluster. Note that the <code>-verbose</code> option will describe why an instance is not compatible with a cluster.
<code>joinCluster -cl cluster_name [-i instance_name]</code>	Add the indicated instance to the specified cluster. An instance is stopped after being added to a cluster and you can manually start it.

Table F-7 Cluster Commands and Their Options (Cont.)

Command	Description
leaveCluster [-i <i>instance_name</i>]	Remove the indicated instance from the cluster. An instance is stopped after being removed from a cluster and you can manually start it.
updateConfig [-ct <i>type</i>]	Update the DCM repository and other members of the cluster with configuration changes made by manually editing the component configuration files. The <i>componentType</i> indicates the component type that has been edited (ohs, oc4j, or opmn). The default is all component types.
resyncInstance [-force] [-i <i>instance_name</i>]	Resynchronize an application server instance with other instances in the cluster. This command can be used after a synchronization operation failed. For example, if you deployed an application across a cluster, and one instance was not able to deploy the application due to insufficient disk space, you could correct the disk space problem and run this command to redeploy the application across all instances. The <i>-force</i> option causes an instance to resynchronize with information from the DCM repository regardless of whether the instance state indicates that it requires resynchronization.

Deploying Applications

This section describes commands for deploying, redeploying, and undeploying OC4J applications.

On hosts with multiple OC4J instances, `dcmctl` determines the target OC4J instance as follows:

- If an OC4J instance is specified with the `-co` target option, apply the operation to that OC4J instance within the associated application server instance. The application server instance is determined first by the `-oraclehome` option, and second by the Oracle home directory in which the `dcmctl` executable resides. If the application server instance is part of a cluster, apply the operation to all OC4J instances with the specified name within the cluster.
- If the `-co` target option is not supplied, apply the operation to all OC4J instances within the associated application server instance. The application server instance is determined first by the `-oraclehome` option, and second by the Oracle home directory in which the `dcmctl` executable resides. If the application server instance is part of a cluster, apply the operation to all OC4J instances within the cluster.

Table F-8 Deployment Commands and Their Options

Command	Description
<code>deployApplication -file name -a app_name [-co comp_name] [-rc root_context]</code>	Deploy an application to the current instance using the WAR or EAR file supplied with the <code>-file</code> option. The application name is assigned to the application for administrative purposes. The name used to access the application from the Web is still the name supplied in the EAR file. The <code>-rc</code> option is required if the application is a WAR file. Do not use the <code>-rc</code> option when deploying an EAR file
<code>redeployApplication -file name -a app_name [-co comp_name] [-rc root_context]</code>	Redeploy an application to the current instance using the WAR or EAR file indicated by the <code>-file</code> option and associate the indicated name as the name of the application for administrative purposes. The <code>-rc</code> option is required if the application is a WAR file. Do not use the <code>-rc</code> option when deploying an EAR file

Table F-8 Deployment Commands and Their Options (Cont.)

Command	Description
<code>undeployApplication -a app_name [-co comp_name]</code>	Undeploy the indicated application.
<code>listApplications -co comp_name [[-cl cluster_name] [-i instance_name]]</code>	Return a list of the applications deployed within the indicated OC4J component. Note that this command allows you to specify an instance or cluster that contains the OC4J component.
<code>validateEarFile -file simple_ear_file</code>	Determine if the supplied EAR file is J2EE compliant. In order to run this command, you must set up your proxy so that Document Type Definitions (DTDs) may be reached on the Web. See Also: " Passing Parameters to the JVM " on page F-6 for more information.

Saving a Backup

Table F-9 lists commands that you can use to back up your application instance, including clustering information, configuration, and applications deployed.

Table F-9 Backup Commands and Their Options

Command	Description
<code>saveInstance -dir directory_name</code>	Save the configuration and application information of the current instance to the designated directory. Create the directory if it does not exist. If it does exit, then the specified directory must be empty. This command can be used to save current configuration settings and installed J2EE applications before making configuration changes. You can then back out of the changes, if necessary, using the <code>restoreInstance</code> command.
<code>restoreInstance [-dir directory_name]</code>	Restore the configuration and application information from the specified directory for this instance. If no directory is specified, then the instance is restored to the configuration set at install time. This command causes the instance to be shut down. If the instance is a member of a cluster, it is removed from the cluster before the information is restored. <code>RestoreInstance</code> does not effect the configuration of the other members of the cluster.
<code>resetFileTransaction</code>	When using a file-based repository for your application instance, it may leave uncommitted information in the repository if an operation is interrupted (control-C). This command blocks all subsequent updates to the repository, cleans up uncommitted data, and reopens the repository for update.

Using the dcmctl Shell

You can execute `dcmctl` commands from within the `dcmctl` shell. To start the `dcmctl` shell, type:

```
dcmctl shell
```

The following is a sample session using the `dcmctl` shell:

```
dcmctl shell
dcmctl> createcluster testcluster
dcmctl> joincluster testcluster
dcmctl> createcomponent -ct oc4j -co component1
dcmctl> start -co component1
dcmctl> deployapplication -f /stage/apps/app1.ear -a app1 -co component1
dcmctl> start -cl testcluster
dcmctl> getstate
dcmctl> exit
```

Note: You can repeat any command within the `dcmctl` shell using the `!!` command.

Executing `dcmctl` from a Command File

You can execute `dcmctl` commands from a script file using the following command:

```
dcmctl shell -f script_file_name
```

For example, create a file called `testFile.cmd` containing the following lines:

```
# this is a comment in a dcmctl command file
echo "creating testcluster"
createcluster testcluster
echo "joining testcluster"
joincluster testcluster
echo "creating component component1"
createcomponent -ct oc4j -co component1
echo "starting component to deploy application"
start -co component1
echo " deploying application"
```

```
deployapplication -f /stage/apps/app1.ear -a app1 -co component1
echo "starting the cluster"
start -cl testcluster
echo "verifying everything started "
getstate
exit
```

Execute `testFile.cmd` using the following command:

```
dcmctl shell -f testFile.cmd
```


G

Enterprise Manager Command-Line Utility (emctl)

The `emctl` utility can be used to start, stop, and manage security for the Enterprise Manager Web site. [Table G-1](#) contains a summary of `emctl` commands.

See Also: [Chapter 3, "Introduction to Administration Tools"](#) for more information about `emctl`

The `emctl` command is located at:

(UNIX) `ORACLE_HOME/bin/emctl`
(Windows) `ORACLE_HOME\bin\emctl`

Table G–1 emctl Commands and Options

Command	Description
<code>emctl start</code>	Start the Enterprise Manager Web site.
<code>emctl stop</code>	Stop the Enterprise Manager Web site (requires <code>ias_admin</code> password).
<code>emctl status</code>	Verify the status of the Enterprise Manager Web site.
<code>emctl set password <i>new_password</i></code>	Reset the <code>ias_admin</code> password.
<code>emctl authenticate <i>password</i></code>	Verify that the supplied password is the <code>ias_admin</code> password.
<code>emctl ssl test</code>	Enable SSL for the Enterprise Manager Web site in a testing environment.
<code>emctl ssl off</code>	Disable SSL for the Enterprise Manager Web site in a testing environment.
<code>emctl set ssl password <i>old_password</i> <i>new_password</i></code>	Set a password to protect an SSL Server Site Certificate. The initial default password is <code>welcome</code> .
<code>emctl gencertrequest</code>	Generate a request for a new SSL Server Site Certificate
<code>emctl installcert -ca <i>certificate_authority_cert_path</i></code>	Install a Certificate Authority Certificate where <code>certificate_authority_cert_path</code> is the full path to the certificate.
<code>emctl installcert -cert <i>SSL_server_site_cert_path</i></code>	Install an SSL Server Site Certificate where <code>SSL_server_site_cert_path</code> is the full path to the certificate.

Backup and Restore

To ensure that you can make a full recovery from media failures, you should perform regular backups of the following:

- [Application Server and Infrastructure Oracle Homes](#)
- [Oracle Internet Directory](#)
- [Metadata Repository](#)
- [Customer Databases](#)

Application Server and Infrastructure Oracle Homes

You should perform regular backups of all files in the Oracle home of each application server and infrastructure installation in your enterprise using your preferred method of filesystem backup.

Oracle Internet Directory

Oracle Internet Directory offers command-line tools for backing up and restoring the Oracle Internet Directory schema and subtree.

See Also: *Oracle Internet Directory Administrator's Guide* for information on backing up and restoring Oracle Internet Directory

Metadata Repository

The metadata repository is an Oracle9*i* Enterprise Edition Database that you can back up and restore using several different tools and operating system commands.

See Also: *Oracle9*i* Backup and Recovery Concepts* for an introduction to Oracle9*i* backup and recovery strategies and tools

Customer Databases

You should perform regular backups of each customer database using the tools available for that particular version of the database.

See Also: Database documentation for the version of each customer database.

Checking Oracle9iAS Version Numbers

You can view Oracle9iAS version numbers in the following ways:

- Launch Oracle Universal Installer by executing
ORACLE_HOME/bin/runInstaller, and click **Installed Products**.
- On Unix, you can also enter the following commands:

```
cd $ORACLE_HOME/inventory  
ls -d Components*/*/*
```

Index

A

access file log, 6-7
access log properties
 changing, 6-9
access.conf, E-3
active EJB methods, 9-9, 9-11
active requests, 9-8, 9-10
active sessions, 9-8, 9-10
active transactions, 9-8
adding
 access file log, 6-7
additional documentation, 1-6
administering users, 12-11
 in a new Oracle Internet Directory, 12-11
 in an existing Oracle Internet Directory, 12-11
administration
 concepts, 2-1
 infrastructure, 2-7
 overview, 2-2
 introduction, 1-1, 7-1
 tasks for Oracle9iAS administrator, 1-2, 4-2
administrative privileges, 12-16
administrator email, 6-4
admin.jar tool
 undeployment, 8-13
agent.xml, E-5
AJP
 port numbers, D-2
AJP port, 13-9
ApJServGroup, 6-14
ApJServGroupMount, 6-15
ApJServGroupSecretKey, 6-15
ApJServManual, 6-14

apologies pages served, 9-13
application
 example, 8-10
 undeployment, 8-13
application server
 administration
 overview, 1-1
 priorities, 1-2
 backup and restore, H-2
 clustering, 14-1
 install types, 2-3, 2-4
 installation, 2-3
 installing
 infrastructure, 2-9
 restarting, 5-4
 starting, 5-4
 stopping, 5-4
application server enterprise
 designing, 2-18
 installing, 2-21
 coexistence with other products, 2-21
 oracle home, 2-21
managing, 2-24
 application server instances, 2-24
 clusters, 2-29
 component instances, 2-26
 farms, 2-28
 reconfiguring, 2-29
applications
 deploying, 8-1
AsynchNotification.properties, E-4
AsynchRequest.properties, E-4
auto-restart, 13-4
availability, 14-3

B

backup and restore, H-1
application server and infrastructure oracle homes, H-2
customer databases, H-2
metadata repository, H-2
Oracle Internet Directory, H-2
BC4J
starting, 5-12
stopping, 5-12
BLEWIS, C-4

C

cache clusters, 9-13, 13-4
cache.conf, E-3
CDOUGLAS, C-4
cgicmd.dat, E-5
changing
access log properties, 6-9
error log properties, 6-6
click-apache.conf, E-5
click-app.properties, E-5
client request, 6-9
CLKANA, C-2
cluster configuration, 14-26
clustered environment
dcmctl, F-5
clustering, 14-1, 14-1 to 14-39
application server instance
adding, 14-11
role, 14-9
architecture, 14-7
benefits, 14-2
availability, 14-3
manageability, 14-4
scalability, 14-2
cluster configuration, 14-26
component instances, 14-14
OC4J instance, 14-15
OC4J process, 14-17
Oracle HTTP Server, 14-14
configuration, 14-26
stateful session bean replication, 14-34

configure replication, 14-32
configuring
EJB application state replication, 14-33
EJB replication configuration in application JAR, 14-34
entity bean replication, 14-34
instance-specific parameters, 14-37
islands, 14-31
Oracle9iAS Single Sign-On, 14-35
processes, 14-31
Web application state replication, 14-32
configuring islands, 14-31
configuring OC4J instance, 14-15
configuring OC4J processes, 14-31
creating a cluster, 14-26
DCM, 14-12
deleting a cluster, 14-28
EJB applications, 14-19, 14-33
hardware load balancer, 14-8
instance-specific parameters, 14-26
OC4J, 14-37, 14-38
Oracle HTTP Server, 14-38
islands, 14-18
J2EE applications, 14-20
managed, 14-2
management features, 14-12
managing
application server instances in a cluster, 14-29
Oracle9iAS clusters, 14-26
non-managed, 14-2
OC4J instance, 14-17
OC4J instance configuration, 14-31
Oracle Enterprise Manager, 14-4
Oracle Enterprise Manager configuration tree, 14-21
Oracle HTTP Server role, 14-15
Oracle9iAS Web Cache, 14-10
overview, 14-2
removing application server instance, 14-30
replicating application state, 14-18
SSO, 14-35
state replication request, 14-20
tuning parameters, 14-22, 14-37
using infrastructure, 14-9

Web applications, 14-19, 14-32
clusters, 2-29
collector-agent.properties, E-5
component configuration dependencies, 7-1
 metadata repository, 7-24
OPMN, 7-10
Oracle HTTP Server, 7-2
Oracle Internet Directory, 7-21
Oracle9i AS Single Sign-On, 7-17
Portal, 7-11
provider groups, 7-15
Web Cache, 7-7
Web providers, 7-13
Wireless, 7-16
components
 restarting, 5-9
 starting, 5-9
 stopping, 5-9
config-file, 13-8, 13-9
 path, 13-8
 start-mode, 13-8
configuration
 OC4J, 6-21
configuration files, E-1
 JServ, E-3
 jserv.conf, E-3
 jserv.properties, E-3
 zone.properties, E-3
 mod_oradav, E-3
 moddav.conf, E-3
 mod_plsql, E-3
 cache.conf, E-3
 dad.conf, E-3
OPMN, E-6
Oracle HTTP Server, E-3
 access.conf, E-3
 httpd.conf, E-3
 magic, E-3
 mime.types, E-3
 mod_oc4j.conf, E-3
 oracle_apache.conf, E-3
 srm.conf, E-3
Oracle Process Manager and Notification
 ons.conf, E-6
 opmn.xml, E-6

Oracle9iAS Clickstream Intelligence, E-5
 agent.xml, E-5
 click-apache.conf, E-5
 click-app.properties, E-5
 collector-agant.properties, E-5
Oracle9iAS Discoverer, E-5
 configuration.xml, E-5
 plus_config.xml, E-5
 portal_config.xml, E-5
 pref.txt, E-5
 .reg_key.dc, E-5
 viewer_config.xml, E-5
Oracle9iAS Forms Services, E-5
 default.env, E-5
 formsweb.cfg, E-5
Oracle9iAS Personalization, E-6
 web.xml, E-6
Oracle9iAS Portal, E-4
 oracle_portal.xml, E-4
 oracle_sso_server.xml, E-4
 targets.xml, E-4
Oracle9iAS Reports Services, E-5
 cgicmd.dat, E-5
 jdbcpds.conf, E-5
 proxyinfo.xml, E-5
 reports_server_name.conf, E-5
 rwbuilder.conf, E-5
 rwserver.template, E-5
 rwervlet.properties, E-5
 textpds.conf, E-5
 xmlpds.conf, E-5
Oracle9iAS Single Sign-On, E-6
 dad.conf, E-6
 plsql.conf, E-6
Oracle9iAS Unified Messaging, E-6
 DBToLDAPMigrationTool.properties, E-6
 mod_osso.conf, E-6
 oc4j.properties, E-6
 plsql.conf, E-6
 toolkit.properties, E-6
Oracle9iAS Web Cache, E-3
 internal_admin.xml, E-3
 internal.xml, E-3
 webcache.xml, E-3

Oracle9iAS Wireless
AsynchNotification.properties, E-4
AsynchRequest.properties, E-4
Cookies.properties, E-4
EncodingSets.properties, E-4
Ftp.properties, E-4
LocalToCharsetMap.properties, E-4
LocationMark_en_US.properties, E-4
LocationMark.properties, E-4
MimeTypes.properties, E-4
Notification.properties, E-4
persistent.properties, E-4
Provisioning.properties, E-4
ProxyFirewall.properties, E-4
System.properties, E-4
System.properties.templ, E-4
UserAgents.properties, E-4
WebBeanProperties.properties, E-4
XMLXSL.properties, E-4
YPCategories.xml, E-4
server.xml file, 8-5
configuration.xml, E-5
configuring, 6-1
infrastructure, 2-9
OC4J
 manual editing, 6-2
Oracle HTTP Server, 6-3
 adding access file log, 6-7
 changing access log properties, 6-9
 changing error log properties, 6-6
 connection handling, 6-9
 creating DAD, 6-11
 editing server configuration files, 6-10
 managing client request, 6-9
 manual editing, 6-2
 modifying document root, 6-4
 modifying group, 6-4
 modifying user, 6-4
 specifying listener port, 6-5
Oracle Management Server, 3-21
Oracle9iAS Web Cache, 6-23
 overview, 6-2
connection handling, 6-9
connection metrics, 9-4
connection rerouting, 13-3
Cookies.properties, E-4
CPU, B-2
 usage, 9-8, 9-13
create users, 12-11
createCluster, F-10
createComponent, F-9
creating
 Database Access Descriptor, 6-11
 OC4J instance, 6-21
custom, 13-11
 environment, 13-11
 gid, 13-11
 maxRetry, 13-11
 num_of_proc, 13-11
 prop, 13-12
 start, 13-11
custom user manager, 8-8
customer database, 2-11
 backup and restore, H-2

D

dad.conf, E-3, E-6
default.env, E-5
data served, 9-13
Database Access Descriptor
 creating, 6-11
DBToLDAPMigrationTool.properties, E-6
DCM, 14-12, C-2
 clustering, 14-13
 command-line utility, F-1
 metadata repository schemas, C-2
 overview, F-1
 role in clustering, 14-12
dcmctl, 14-13, F-1
 a app_name, F-4
 administration, F-7
 application, F-4
 backup, F-13
 cluster, F-4
 cluster capabilities, F-5
 clustered environment, F-5
 command overview, F-3

commands and options, F-3
 target options, F-3
 universal options, F-4
component, F-4
componentType, F-4
createCluster, F-10
createComponent, F-9
DCM utility, F-1
debug, F-4
deployApplication, F-12
deploying applications, F-12
destroyInstance, F-8
executing from command file, F-14
execution from a file, F-14
execution from a shell, F-14
getComponentTypes, F-9
getError, F-2
getError command, F-2
getReturnStatus, F-2
getstart, F-7
help, F-2
help command, F-2
instance, F-4
isClusterable, F-10
isCompatible, F-10
joinCluster, F-10
leaveCluster, F-11
listApplications, F-13
listClusters, F-10
listComponents, F-8
listComponentTypes, F-9
listInstances, F-8, F-10
logdir, F-4
managing
 application server instances, F-8
 clusters, F-10
 components, F-9
managing clusters, F-10
managing components, F-9
managing instances, F-8
oraclehome, F-4
overview, F-2
passing parameters, F-6
passing parameters to JVM, F-6
redeployApplication, F-12
removeCluster, F-10
removeComponent, F-9
resetFileTransaction, F-13
restart, F-7
restoreInstance, F-13
resyncInstance, F-8, F-11
saveInstance, F-13
saving a backup, F-13
shutdown, F-7
start, F-7
starting, F-7
stop, F-7
stopping, F-7
target option list, F-3
timeout, F-4
undeployApplication, F-13
universal option list, F-4
updateConfig, F-9, F-11
using dcmctl shell, F-14
validateEarFile, F-13
verbose, F-5
whichCluster, F-10
whichInstance, F-8
dcmctl shell, F-14
death detection, 13-2
debug, F-4
debugging
 OC4J, 14-38
delegated administrative service, 12-8
delete users, 12-11
deleting
 OC4J instance, 6-22
deploy EAR file, 8-2
deploy WAR file, 8-2
deployApplication, F-12
deploying
 applications, 8-1
 J2EE applications, 8-2
 basic deployment, 8-3
 Petstore Web Application Demo, 8-10
 recovering from deployment errors, 8-13
deployment
 error recovery, 8-13
 deployment errors, 8-13

designing
application server enterprise, 2-18

destroyInstance, F-8

developer kits

 memory requirement, B-3

directives

 ApJServGroup, 6-14

 ApJServGroupMount, 6-15

 ApJServGroupSercetKey, 6-15

 ApJServManual, 6-14

 port, 6-13

Discoverer

 starting, 5-12

 stopping, 5-12

DISCOVERER5, C-2

discovering targets, 3-26

disk space, B-2

DISPLAY, 3-9

Distributed Configuration Management, 14-12

Distributed Configuration Management, see DCM

document root, 6-4

DSGATEWAY, C-3

DSSYS, C-3

Dynamic Configuration Management, 14-7

dynamically generated content, 13-4

E

EAR file

 creation, 8-3

 structure, 8-3

 used in deployment, 8-3

edit user properties, 12-11

editing

 server configuration files, 6-10

EJB

 deployment, 8-3

 replication, 14-34

EJB application state replication, 14-33

EJB replication configuration in application

 JAR, 14-34

emctl, 3-10, G-1

 authenticate password, G-2

 gencertrequest, G-2

 installcert, G-2

set password, G-2

set ssl password, G-2

ssl off, G-2

ssl test, G-2

start, 3-10, G-2

status, 3-10, G-2

stop, 3-10, G-2

EncodingSets.properties, E-4

end of call replication, 14-34

Enterprise JavaBeans, see EJB

entity bean replication

 configuring, 14-34

error log properties

 changing, 6-6

error metrics, 9-4

F

failover, 13-2, 13-3

farms, 2-28

fault tolerance, 13-2

files

 configuration, E-1

 log, E-1

Forms

 starting, 5-12

 stopping, 5-12

formsweb.cfg, E-5

Ftp.properties, E-4

G

getComponentTypes, F-9

getError, F-2

getReturnStatus, F-2

getstart, F-7

gid, 13-7, 13-8, 13-11

group, 6-5

H

hardware load balancer, 14-8
hardware requirements, B-2
 CPU, B-2
 disk space, B-2
 monitor, B-2
 swap space, B-2
 TMP, B-2
heap
 usage, 9-8
high availability, 13-1
 architecture, 13-5
 benefits, 13-2
 connection rerouting, 13-3
 death detection and restart, 13-2
 failover, 13-2
 fault tolerance, 13-2
failure scenarios, 13-17
 OC4J, 13-19
 OPMN, 13-20
 Oracle HTTP Server, 13-17
introduction, 13-2
Oracle9iAS Web Cache, 13-3
 auto-restart, 13-4
 cache clusters, 13-4
 dynamically generated content, 13-4
 failover, 13-3
 load balancing, 13-3
 surge protection, 13-4
httpd.conf, E-3

I

IIOP
 port numbers, D-2
inCompatible, F-10
Infrastructure
 memory requirement, B-3
infrastructure, 2-7
 basic administration, 11-2
 backing up and restoring, 11-3
 starting and stopping, 11-2
 managing, 11-1
 instance, 11-3

metadata repository, 11-7
Oracle Internet Directory, 11-7
Oracle9iAS Single Sign-On, 11-4
metadata repository
 changing password, 11-8
 installation and configuration, 11-7
 starting and stopping, 11-8
Oracle Internet Directory
 starting and stopping, 11-12
Oracle9iAS Single Sign-On
 accessing single sign-on server, 11-6
 administrators, 11-5
 installation and configuration, 11-5
 multilingual support, 11-6
 starting, 5-20, 5-23
infrastructure oracle homes
 backup and restore, H-2
install types, 2-3
 Business Intelligence and Forms, 2-4
 J2EE and Web Cache, 2-4
 Portal and Wireless, 2-4
 Unified Messaging, 2-4
installation
 application server, 2-3
installing
 application server enterprise, 2-21
 Intelligent Agent, 3-19
 Oracle Enterprise Manager Console, 3-20
 Oracle Management Server, 3-20
instanceName, 13-8
instance-specific parameters, 14-26
Intelligent Agent
 installing, 3-19
 starting, 3-21
 stopping, 3-21
internal_admin.xml, E-3
internal.xml, E-3
Internet File System
 starting, 5-12
 stopping, 5-12
isClusterable, F-10
island, 13-9
islands, 14-18

J

J2EE and Web Cache
 memory requirement, B-3
 restarting, 5-7
 starting, 5-7
 stopping, 5-7
J2EE applications
 deployment, 8-2
 undeploying, 8-13
java object cache
 port numbers, D-3
java-bin, 13-9
java-option, 13-9
JAZN LDAP user manager, 8-7
JAZN xml user manager, 8-7
jdbcpds.conf, E-5
JMS
 port numbers, D-2
JMS port, 13-9
joinCluster, F-10
JServ
 automatic mode, 6-17
 coexistence with OC4J, 6-19
 configuration files, E-3
 directive reference, 6-13
 jserv.conf, 6-14
 jserv.properties, 6-13
 enabling, 6-12
 manual mode, 6-18
 mod_oprocmgr, 6-16
 OC4J, 6-19
 starting, 5-13
 stopping, 5-13
jserv.conf, E-3
 port, 6-14, 6-15
jserv.properties, E-3
 port, 6-13
JSP pages
 deployment, 8-3
JVM, F-6

K

KWALKER, C-4

L

leaveCluster, F-11
listApplications, F-13
listClusters, F-10
listComponents, F-8
listComponentTypes, F-9
listener port
 specifying, 6-5
listInstances, F-8, F-10
load balancing, 13-3
LocalToCharsetMap.properties, E-4
LocationMark_en_US.properties, E-4
LocationMark.properties, E-4
log files, E-1
logdir, F-4
log-file, 13-7, 13-16

M

magic, E-3
manageability, 14-4
managing
 application server enterprise, 2-24
 application server instances in a cluster, 14-29
 adding, 14-29
 removing, 14-30
 infrastructure, 11-1
 Oracle9iAS clusters, 14-26
 Oracle9iAS Single Sign-On, 11-4
 security, 12-1
manifest file, 8-3
maxRetry, 13-7, 13-8, 13-11
memory
 usage, 9-8, 9-13
memory requirements, B-3
 developer kits, B-3
 Infrastructure, B-3
 J2EE and Web Cache, B-3
metadata repository, 2-7
 backup and restore, H-2
 changing password, 11-8
 component configuration dependencies, 7-24
 installation and configuration, 11-7
 managing, 11-7

schemas, C-1
 starting and stopping, 11-8
 metadata repository schemas
 DCM, C-2
 Oracle Certificate Manager, C-2
 OCM_DB_ADMIN, C-2
 Oracle Enterprise Manager, C-3
 OEM_REPOSITORY, C-3
 Oracle Internet Directory, C-3
 ODS, C-3
 ODSCOMMON, C-3
 Oracle Workflow, C-4
 Oracle9iAS Clickstream Intelligence, C-2
 CLKANA, C-2
 CLKRT, C-2
 Oracle9iAS Discoverer, C-2
 DISCOVERER5, C-2
 Oracle9iAS InterConnect, C-2
 OAIHUB9020, C-2
 Oracle9iAS Portal, C-3
 PORTAL, C-3
 PORTAL_APP, C-3
 PORTAL_DEMO, C-3
 PORTAL_PUBLIC, C-3
 Oracle9iAS Single Sign-On, C-3
 ORASSO, C-3
 ORASSO_PA, C-3
 ORASSO_PUBLIC, C-3
 Oracle9iAS Syndication Server, C-3
 DSGATEWAY, C-3
 Oracle9iAS Web Services, C-3
 DSSYS, C-3
 UDDISYS, C-3
 Oracle9iAS Wireless, C-4
 BLEWIS, C-4
 CDOUGLAS, C-4
 KWALKER, C-4
 OWF_MGR, C-4
 SYSADMIN, C-4
 WFADMIN, C-4
 SCOTT, C-2
 SYS, C-2
 SYSTEM, C-2
 method execution rate, 9-9, 9-11
 metrics

 connection, 9-4
 error, 9-4
 system usage, 9-3
 mime.types, E-3
 MimeTypes.properties, E-4
 mod_oc4j, 13-5
 mod_oc4j.conf, E-3
 mod_oprocmgr, 6-16
 mod_jserv, 6-16
 mod_oradav
 configuration files, E-3
 mod_osso.conf, E-6
 mod_plsql
 configuration files, E-3
 moddav.conf, E-3
 modifying
 document root setting, 6-4
 group setting, 6-4
 user settings, 6-4
 monitoring
 OC4J, 9-1, 9-6
 application pages, 9-9
 OC4J Home Page, 9-6
 Oracle HTTP Server, 9-1, 9-2
 metrics page, 9-3
 module metrics page, 9-5
 Oracle HTTP Server Home Page, 9-2
 response and load metrics page, 9-5
 Oracle9iAS Web Cache, 9-1, 9-11
 activity page, 9-14
 application Web servers page, 9-16
 Oracle9iAS Web Cache Home Page, 9-12
 popular cached documents page, 9-18

N

Notification.properties, E-4
 num_of_proc, 13-11
 number of cached documents, 9-13
 numProcs, 13-8

O

OAIHUB9020, C-2

OC4J

- application example, 8-10
- clustering role, 14-17
- coexistence with JServ, 6-19
- configuration, 6-21
- configuration tasks, 6-22
- creating instance, 6-21
- debugging, 14-38
- demos, 4-8
- instance
 - deleting, 6-22
- JServ, 6-19
- port numbers, D-2

oc4j, 13-8

- config-file, 13-9
- gid, 13-8
- instanceName, 13-8
- island, 13-9
- java-bin, 13-9
- java-option, 13-9
- maxRetry, 13-8
- numProcs, 13-8
- oc4j-option, 13-9
- port, 13-9
- prop, 13-10
- timeout, 13-8

OC4J instance, 14-15

- configuring
 - clustering, 14-31

OC4J instance-specific parameters, 14-37, 14-38

OC4J process, 14-17

oc4j-option, 13-9

oc4j.properties, E-6

OC4J-ready Pet Store Demo, 8-10

OCM_DB_ADMIN, C-2

ODS, C-3

ODSCOMMON, C-3

OEM_REPOSITORY, C-3

OEM_RESPOSITORY, 11-9

ohs, 13-7

- gid, 13-7
- maxRetry, 13-7

timeout, 13-8

ons.conf, E-6

open connections, 9-13

JDBC, 9-8

OPMN, 13-6, 14-13

component configuration dependencies, 7-10

configuration files, E-6

Oracle Notification System, 13-6

Oracle Process Manager, 13-6

role in clustering, 14-13

opmn.xml, E-6

Oracle Certificate Manager

metadata repository schemas, C-2

Oracle Directory Manager, 12-8

Oracle Enterprise Manager, A-4

clustering

configuration tree, 14-21

configuration assistant, 3-22

starting, 3-23

Console, 3-18, A-4

metadata repository schemas, C-3

port numbers, D-4

three-tier framework, 3-19

installing, 3-19

Web site, A-4

Oracle Enterprise Manager Console, 3-18, A-4

installing, 3-20

starting, 3-25

using, 3-25

Oracle Enterprise Manager Web site, 3-2, A-4

introduction, 3-3

locating the UNIX process ID, 3-11

security, 3-13

using SSL, 3-13, 3-14

starting, 3-9

stopping, 3-9, 3-10, 3-11

using console, 3-28

viewing, 3-4

Oracle HTTP Server, A-4

clustering role, 14-15

component configuration dependencies, 7-2

configuration files, E-3

connection handling, 6-9

creating DAD, 6-11

editing
server configuration file, 6-10
managing client request, 6-9
port numbers, D-3

Oracle HTTP Server instance-specific parameters, 14-38

Oracle Internet Directory, 2-7, A-4
adding user to group, 12-12
administering groups, 12-13
in a new Oracle Internet Directory, 12-13
in an existing Oracle Internet Directory, 12-14
administering users, 12-11
administrative privileges, 12-16
backup and restore, H-2
changing component schema user passwords, 10-5
changing instance password, 12-21
component configuration dependencies, 7-21
creating user, 12-11
delegating administration tasks, 12-9
determining group's privileges, 12-15
existing, 12-3
group details, 12-14
ias_admin user, 12-5
in Oracle9iAS, 12-3
installing and configuring Oracle9iAS, 12-16
managing, 11-7
metadata repository schemas, C-3
modifying password policies, 12-18
 modifying, 12-19
 password policies, 12-18
 viewing, 12-18
modifying user passwords, 12-20
 modify orcladmin password, 12-20
new and existing instances, 12-3
newly created, 12-3
port numbers, D-5
starting and stopping, 11-12
superuser and administrators, 12-4
tasks, 12-6
tools, 12-7
upgrading, 12-17

Oracle Management Server, 2-7
configuring, 3-21

installing, 3-20
port numbers, D-4
starting, 3-24
stopping, 3-24

Oracle Notification System, 13-6
 configuring, 13-15
 sample configurations, 13-16

Oracle Process Management and Notification, 14-7, 14-13

Oracle Process Management Notification, see OPMN

Oracle Process Manager, 13-6
 configuring, 13-7
 process-manager, 13-7
 sample configurations, 13-12

Oracle Workflow
 metadata repository schemas, C-4
 port numbers, D-5
 oracle_apache.conf, E-3
 oracle_portal.xml, E-4
 oracle_sso_server.xml, E-4

Oracle9i Application Server components
 Oracle Enterprise Manager, A-4
 Oracle HTTP Server, A-4
 Oracle Internet Directory, A-4
 Oracle9iAS Clickstream Intelligence, A-1
 Oracle9iAS Containers for J2EE, A-1
 Oracle9iAS Discoverer, A-1
 Oracle9iAS Forms Services, A-2, A-3
 Oracle9iAS InterConnect, A-2
 Oracle9iAS Personalization, A-2, A-3
 Oracle9iAS Portal, A-2
 Oracle9iAS Reports Services, A-2
 Oracle9iAS Unified Messaging, A-3
 Oracle9iAS Web Cache, A-3
 Oracle9iAS Web Services, A-3
 Oracle9iAS Wireless, A-3

configuring
 install components metadata into customer database, 4-4
 unconfigure and configure components, 4-4

version numbers, I-1

Oracle9i AS Single Sign-On
 component configuration dependencies, 7-17

Oracle9iAS administration
 documentation, 4-2
 tools, 3-1

Oracle9iAS administrative tools
 Oracle Enterprise Manager Web site, 3-2

Oracle9iAS Clickstream Collector Server
 port numbers, D-2

Oracle9iAS Clickstream Execution Engine
 port numbers, D-2

Oracle9iAS Clickstream Intelligence, A-1
 configuration files, E-5
 metadata repository schemas, C-2
 port numbers, D-2

Oracle9iAS Clickstream Intelligence Viewer
 port numbers, D-2

Oracle9iAS Clickstream Runtime Administrator
 port numbers, D-2

Oracle9iAS Component Home Page, 3-8

Oracle9iAS components
 restarting, 5-9
 starting, 5-9
 stopping, 5-9

Oracle9iAS Containers for J2EE, A-1

Oracle9iAS Developer Kits
 demos, 4-8

Oracle9iAS Discoverer, A-1
 configuration files, E-5
 demos, 4-9
 metadata repository schemas, C-2
 port numbers, D-2

Oracle9iAS Farm Home Page, 3-7

Oracle9iAS Forms Services, A-2
 configuration files, E-5
 demos, 4-9
 port numbers, D-2

Oracle9iAS Instance Home Page, 3-5

Oracle9iAS InterConnect, A-2
 metadata repository schemas, C-2

Oracle9iAS Personalization, A-2, A-3
 configuration files, E-6

Oracle9iAS Portal, A-2
 configuration files, E-4
 demos, 4-9
 metadata repository schemas, C-3
 port numbers, D-3

Oracle9iAS Reports Services, A-2
 configuration files, E-5
 demos, 4-9
 port numbers, D-3

Oracle9iAS Single Sign-On, 2-7, A-3
 accessing single sign-on server, 11-6
 administrators, 11-5
 clustering, 14-35
 configuration files, E-6
 installation and configuration, 11-5
 managing, 11-4
 metadata repository schemas, C-3
 multilingual support, 11-6
 port numbers, D-3

Oracle9iAS Syndication Server
 metadata repository schemas, C-3

Oracle9iAS Unified Messaging, A-3
 configuration files, E-6
 port numbers, D-4
 IMAP4, D-4
 IMAP4-SSL, D-4
 NNTP, D-4
 NNTP-SSL, D-4
 POP3, D-4
 POP3-SSL, D-4
 SMTP, D-4

Oracle9iAS Web Cache, A-3
 clustering, 14-10
 configuration files, E-3
 configuring, 6-23
 demos, 4-8
 memory, B-3
 port numbers, D-4

Oracle9iAS Web Services, A-3
 demos, 4-8
 metadata repository schemas, C-3

Oracle9iAS Welcome Page
 viewing, 4-2

Oracle9iAS Wireless, A-3
 demos, 4-9
 metadata repository schemas, C-4
 port numbers, D-4

oraclehome, F-4

ORASSO, C-3

ORASSO_DS, C-3

ORASSO_PA, C-3
ORASSO_PS, C-3
ORASSO_PUBLIC, C-3
orcladmin, 12-20
out of memory error, 8-13
OWF_MGR, C-4

P

parameters
 instance-specific, 14-26
 OC4J instance-specific, 14-37, 14-38
 Oracle HTTP Server instance-specific, 14-38
path, 13-8
persistent.properties, E-4
Pet Store Web Application demo, 8-10
plsql.conf, E-6
plus_config.xml, E-5
port, 13-9, 13-15
 local, 13-15
 remote, 13-15
 request, 13-15
port allocation, 2-22
port conflicts, 4-3
port numbers, D-1
 IIOP, D-2
 OC4J, D-2
 AJP, D-2
 HTTP listener, D-2
 JMS, D-2
 RMI, D-2
 Oracle Enterprise Manager, D-4
 administration, D-4
 Intelligent Agent, D-4
 Oracle HTTP Server, D-3
 java object cache, D-3
 JServ servlet, D-3
 non-SSL, D-3
 Oracle Notification Service local port, D-3
 Oracle Notification Service remote port, D-3
 Oracle Notification Service report port, D-3
 Oracle Notification Service requested
 port, D-3
 SSL, D-3
 Oracle Internet Directory, D-5

non-SSL, D-5
SSL, D-5
Oracle Internet File System, D-5
 AFP, D-5
 CUP, D-5
 domain controller, D-5
 domain controller and nodes, D-5
 FTP, D-5
 HTTP node, D-5
 IMAP, D-5
 IMAP4-SSL, D-5
 LDAP-non-SSL, D-5
 LDAP-SSL, D-5
 main node, D-5
 NFS, D-5
 servlet, D-5
 SMB, D-5
 SMTP, D-5
 WCP, D-5
Oracle Management Server, D-4
Oracle Workflow, D-5
 TNS, D-5
Oracle9IAS Clickstream Collector Server, D-2
Oracle9IAS Clickstream Execution Engine, D-2
Oracle9IAS Clickstream Intelligence, D-2
Oracle9IAS Clickstream Intelligence
 Viewer, D-2
Oracle9IAS Clickstream Runtime
 Administrator, D-2
Oracle9IAS Discoverer, D-2
Oracle9IAS Forms Services, D-2
Oracle9IAS Portal, D-3
Oracle9IAS Reports Services, D-3
 Visigenics CORBA, D-3
Oracle9IAS Single Sign-On, D-3
Oracle9IAS Unified Messaging, D-4
 IMAP4, D-4
 IMAP4-SSL, D-4
 NNTP, D-4
 NNTP-SSL, D-4
 POP3, D-4
 POP3-SSL, D-4
 SMTP, D-4

Oracle9iAS Web Cache, D-4
 administration, D-4
 HTTP Listen non-SSL, D-4
 HTTP Listen SSL, D-4
Oracle9iAS Wireless, D-4
 integration server, D-4
sorted by component, D-2
sorted by port numbers, D-6
 SQL*Net, D-3
port ranges, D-1
PORTAL, C-3
Portal
 component configuration dependencies, 7-11
 starting, 5-13
 stopping, 5-13
PORTAL_APP, C-3
portal_config.xml, E-5
PORTAL_DEMO, C-3
PORTAL_PUBLIC, C-3
postinstallation tasks, 4-1
 check port conflicts, 4-3
 complete component configuration, 4-4
 run demos, 4-7
 OC4J, 4-8
 Oracle HTTP Server, 4-7
 Oracle9iAS Developer Kits, 4-8
 Oracle9iAS Discoverer, 4-9
 Oracle9iAS Forms Services, 4-9
 Oracle9iAS Portal, 4-9
 Oracle9iAS Reports Services, 4-9
 Oracle9iAS Web Cache, 4-8
 Oracle9iAS Web Services, 4-8
 Oracle9iAS Wireless, 4-9
 viewing Oracle9iAS Welcome Page, 4-2
pref.txt, E-5
process-manager, 13-7
 log-file, 13-7
 oc4j, 13-8
 ohs, 13-7
prop, 13-10, 13-12
provider groups
 component configuration dependencies, 7-15
Provisioning.properties, E-4
ProxyFirewall.properties, E-4
proxyinfo.xml, E-5

R

reconfiguring
 application server enterprise, 2-29
redeployApplication, F-12
.reg_key.dc, E-5
removeCluster, F-10
removeComponent, F-9
replicating application state, 14-18
reports_server_name.conf, E-5
request processing time, 9-8, 9-10
requests per second, 9-8, 9-10, 9-13
resetFileTransaction, F-13
resetIASpasswd.sh, 12-21
restart, F-7
restarting, 5-4
 application server, 5-4
 command line
 J2EE and Web Cache, 5-7
 components, 5-9
 using component home page, 5-9
 using instance home page, 5-9
 Oracle9iAS components, 5-9
restore, H-1
restoreInstance, F-13
resyncInstance, F-8, F-11
RMI
 port numbers, D-2
RMP port, 13-9
run demos, 4-7
rwbuilder.conf, E-5
rwserver.template, E-5
rw servlet.properties, E-5

S

sample application server enterprise designs, 2-13
sample certificate requests, 3-16
sample SSL certificate, 3-17
saveInstance, F-13
scalability, 14-2
schemas
 metadata repository, C-1
SCOTT, 11-9, C-2

security, 12-1
Oracle Enterprise Manager Web site, 3-13
Oracle Internet Directory, 12-2
 adding user to group, 12-12
 administering groups, 12-13
 administering users, 12-11
 administrative privileges, 12-16
 changing component schema user
 passwords, 10-5
 changing instance passwords, 12-21
 creating user, 12-11
 delegating administration tasks, 12-9
 determining group's privileges, 12-15
 group details, 12-14
 ias_admin user, 12-5
 in Oracle9iAS, 12-3
 installing and configuring Oracle9iAS, 12-16
 modifying password policy, 12-18
 modifying user passwords, 12-20
 new and existing instances, 12-3
 subscribe and enterprise, 12-2
 superuser and administrators, 12-4
 tasks, 12-6
 tools, 12-7
 upgrading, 12-17
server configuration files
 editing, 6-10
server.xml file, 8-5
servlet engine process, 6-16
servlets
 deployment, 8-3
setting
 DISPLAY, 3-9
shutdown, F-7
Single Sign-On
 starting, 5-13
 stopping, 5-13
size of cached documents, 9-13
specifying
 listener port, 6-5
SQL*Net
 port numbers, D-3
srm.conf, E-3
ssl, 13-15
 enabled, 13-15
HTTP Server start mode, 13-8
wallet-file, 13-15
wallet-password, 13-15
starting, 5-4
application server, 5-4
BC4J, 5-12
command line
 J2EE and Web Cache, 5-7
 OC4J, 5-10, 5-11
 Oracle HTTP Server, 5-10, 5-11
 Web Cache, 5-10, 5-11
component dependencies, 5-13
Clickstream, 5-14
Clickstream Collector, 5-14
OC4J_BI_Forms, 5-14
OC4J_DAS, 5-14
OC4J_Demo, 5-14
OC4J_home, 5-14
OC4J_iFS, 5-14
OC4J_Portal, 5-14
OC4J_UM, 5-14
OC4J_Wireless, 5-14
Oracle HTTP Server, 5-14
Oracle Internet File System, 5-14
Reports, 5-14
Unified Messaging, 5-14
Web Cache, 5-14
Wireless, 5-14
components, 5-9
 using component home page, 5-9
 using instance home page, 5-9
dcmtcl, F-7
Discoverer, 5-12
Forms, 5-12
infrastructure, 5-20, 5-23
Intelligent Agent, 3-21
Internet File System stopping
 Internet File System, 5-12
JServ, 5-13
Oracle Enterprise Manager Configuration Assistant, 3-23
Oracle Enterprise Manager Console, 3-25
Oracle Enterprise Manager Web site, 3-9
Oracle Management Server, 3-24
Oracle9iAS components, 5-9

Portal stopping
 Portal, 5-13
 Single Sign-On, 5-13
start-mode, 13-8
state replication request, 14-20
stateful session bean replication
 configuring, 14-34
 end of call replication, 14-34
 stateful session context, 14-34
 VM termination replication, 14-34
stateful session context, 14-34
stop, F-7
stopping, 5-4
 application server, 5-4
 BC4J, 5-12
 command line
 J2EE and Web Cache, 5-7
 OC4J, 5-10, 5-11
 Oracle HTTP Server, 5-10, 5-11
 Web Cache, 5-10, 5-11
component dependencies, 5-13
 Clickstream, 5-14
 Clickstream Collector, 5-14
 OC4J_BI_Forms, 5-14
 OC4J_DAS, 5-14
 OC4J_Demo, 5-14
 OC4J_home, 5-14
 OC4J_iFS, 5-14
 OC4J_Portal, 5-14
 OC4J_UM, 5-14
 OC4J_Wireless, 5-14
 Oracle HTTP Server, 5-14
 Oracle Internet File System, 5-14
 Reports, 5-14
 Unified Messaging, 5-14
 Web Cache, 5-14
 Wireless, 5-14
components, 5-9
 using component home page, 5-9
 using instance home page, 5-9
dcmctl, F-7
Discoverer, 5-12
Forms, 5-12
Intelligent Agent, 3-21
JServ, 5-13

Oracle Enterprise Manager Web site, 3-9
Oracle Management Server, 3-24
Oracle9iAS components, 5-9
 Single Sign-On, 5-13
superusers, 12-4
surge protection, 13-4
swap space, B-2
SYS, 11-9, C-2
SYSADMIN, C-4
SYSTEM, 11-9, C-2
system usage metrics, 9-3
System.properties, E-4
System.properties.templ, E-4

T

targets.xml, E-4
textpds.conf, E-5
timeout, 13-8, F-4
TMP, B-2
toolkit.properties, E-6
total JDBC connections, 9-8
transaction commits, 9-8
transaction rollbacks, 9-8

U

UDDISYS, C-3
undateConfig, F-9
undeployApplication, F-13
undeploying
 J2EE applications, 8-13
undeployment, 8-13
underlying technologies, 3-17
updateConfig, F-11
usage
 CPU, 9-8, 9-13
 heap, 9-8
 memory, 9-8, 9-13
use infrastructure wizard, 10-4
user, 6-4
UserAgents.properties, E-4
using
 emctl, 3-10
 jobs and events, 3-27

V

validateEarFile, F-13
verbose, F-5
version numbers, I-1
viewer_config.xml, E-5
Visigenics CORBA
 port numbers, D-3
VM termination replication, 14-34

W

Web
 application deployment, 8-3
Web application state replication, 14-32
Web Cache
 component configuration dependencies, 7-7
Web providers
 component configuration dependencies, 7-13
WebBeanProperties.properties, E-4
webcache.xml, E-3
web.xml, E-6
WFADMIN, C-4
whichCluster, F-10
whichInstance, F-8
Wireless
 component configuration dependencies, 7-16

X

xml user manager, 8-7
xmlpds.conf, E-5
XMLXSL.properties, E-4

Y

YPCategories.xml, E-4

Z

zone.properties, E-3

